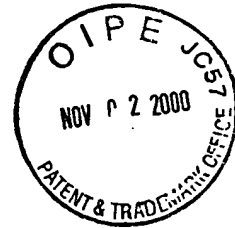


09/605,008
CF0 14575 US/w

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application:

2000年 6月 7日

出 願 番 号
Application Number:

特願2000-170657

出 願 人
Applicant(s):

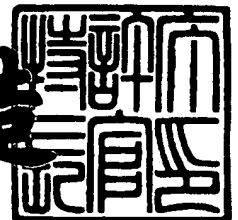
キヤノン株式会社

CERTIFIED COPY OF
PRIORITY DOCUMENT

2000年 7月21日

特 許 庁 長 官
Commissioner,
Patent Office

及 川 耕 造



出証番号 出証特2000-3057159

【書類名】 特許願

【整理番号】 4214011

【提出日】 平成12年 6月 7日

【あて先】 特許庁長官 近藤 隆彦 殿

【国際特許分類】 G06F 3/12

【発明の名称】 情報処理装置、情報処理方法、および記憶媒体

【請求項の数】 33

【発明者】

【住所又は居所】 東京都大田区下丸子3丁目30番2号キヤノン株式会社
内

【氏名】 戸田 雅成

【特許出願人】

【識別番号】 000001007

【住所又は居所】 東京都大田区下丸子3丁目30番2号

【氏名又は名称】 キヤノン株式会社

【代表者】 御手洗 富士夫

【電話番号】 03-3758-2111

【代理人】

【識別番号】 100090538

【住所又は居所】 東京都大田区下丸子3丁目30番2号キヤノン株式会社
内

【弁理士】

【氏名又は名称】 西山 恵三

【電話番号】 03-3758-2111

【選任した代理人】

【識別番号】 100096965

【住所又は居所】 東京都大田区下丸子3丁目30番2号キヤノン株式会
社内

【弁理士】

【氏名又は名称】 内尾 裕一

【電話番号】 03-3758-2111

【選任した代理人】

【識別番号】 100110009

【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号キャノン株式会
社内

【弁理士】

【氏名又は名称】 青木 康

【電話番号】 03-3758-2111

【先の出願に基づく優先権主張】

【出願番号】 平成11年特許願第183332号

【出願日】 平成11年 6月29日

【手数料の表示】

【予納台帳番号】 011224

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9908388

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 情報処理装置、情報処理方法、および記憶媒体

【特許請求の範囲】

【請求項 1】 印刷データをイメージデータに変換する第 1 変換手段と、印刷データをプリンタでイメージデータを生成するコードデータに変換する第 2 変換手段と、前記第 1 変換手段により変換させたイメージデータを用いた印刷時間と前記第 2 変換手段により変換させたコードデータを用いた印刷時間とを予測する予測手段と、前記予測手段による予測結果に基づきイメージデータを用いた印刷時間の方が短いかな否かを判別する判別手段と、前記判別手段によりイメージデータを用いた印刷時間の方が短いと判別された場合、前記第 1 変換手段を選択し、前記判別手段によりイメージデータを用いた印刷時間の方が短くないと判別された場合、前記第 2 変換手段を選択する選択手段とを有することを特徴とする情報処理装置。

【請求項 2】 前記予測手段は、ホストコンピュータ側の処理時間、転送時間、プリンタ側の処理時間に基づき予測することを特徴とする請求項 1 記載の情報処理装置。

【請求項 3】 イメージデータを用いた印刷時間として描画処理時間、転送時間、プリンタの処理時間を含み、コードデータを用いた印刷時間としてコマンド生成時間、サブクローズ時間を含むことを特徴とする請求項 1 記載の情報処理装置。

【請求項 4】 イメージデータを用いた印刷時間として描画処理時間、転送時間、中間データ生成時間、シップ時間を含み、コードデータを用いた印刷時間としてコマンド生成時間、転送時間、シップ準備時間、シップ時間を含むことを特徴とする請求項 1 記載の情報処理装置。

【請求項 5】 前記選択手段は、論理ページ単位で選択することを特徴とする請求項 1 記載の情報処理装置。

【請求項 6】 前記選択手段により選択された変換手段により変換されたデ

ータにより印刷を行なうとプリンタでサブクローズが発生するか否かを判定する判定手段と、

前記判定手段によりサブクローズが発生すると判別された場合、変換手段により変換されるデータの階調を低下させる手段とを有することを特徴とする請求項 1 記載の情報処理装置。

【請求項 7】 印刷データを調べ、描画不正が起こるか否かを判断する判断手段を有し、

前記選択手段は、前記判断手段により描画不正が起こると判断された場合、前記第 1 変換手段を選択し、前記判断手段により描画不正が起きないと判断された場合、前記第 2 変換手段を選択することを特徴とする請求項 1 記載の情報処理装置。

【請求項 8】 印刷データをイメージデータに変換する第 3 変換手段と、印刷データを調べ、描画不正が起こるか否かを判断する判断手段とを有し、前記判断手段により描画不正が起きないと判断された場合、コードデータで処理する時間がイメージデータで処理する時間より長いかな否かを判定する判定手段とを有し、

前記選択手段は、前記判定手段により、コードデータで処理する時間がイメージデータで処理する時間より長いと判定された場合、前記第 3 変換手段を選択し、前記判定手段により、コードデータで処理する時間がイメージデータで処理する時間より長くないと判定された場合、前記第 2 変換手段を選択することを特徴とする請求項 1 記載の情報処理装置。

【請求項 9】 前記第 1 変換手段は RGB で論理演算を行ない、前記第 3 変換手段は YMCK で論理演算を行なうことを特徴とする請求項 8 記載の情報処理装置。

【請求項 10】 印刷データをイメージデータに変換する第 1 変換手段と、印刷データをプリンタでイメージデータを生成するコードデータに変換する第 2 変換手段と、

前記第 1 変換手段および前記第 2 変換手段のいずれかを論理ページ単位で選択する選択手段とを有することを特徴とする情報処理装置。

【請求項 11】 印刷データをイメージデータに変換する第 1 変換手段と、

印刷データをプリンタでイメージデータを生成するコードデータに変換する第 2 変換手段とを利用する情報処理方法であって、

前記第 1 変換手段により変換させたイメージデータを用いた印刷時間と前記第 2 変換手段により変換させたコードデータを用いた印刷時間とを予測する予測ステップと、

前記予測ステップによる予測結果に基づきイメージデータを用いた印刷時間の方が短いかな否かを判別する判別ステップと、

前記判別ステップによりイメージデータを用いた印刷時間の方が短いと判別された場合、前記第 1 変換手段を選択し、前記判別ステップによりイメージデータを用いた印刷時間の方が短くないと判別された場合、前記第 2 変換手段を選択する選択ステップとを有することを特徴とする情報処理方法。

【請求項 1 2】 前記予測ステップは、ホストコンピュータ側の処理時間、転送時間、プリンタ側の処理時間に基づき予測することを特徴とする請求項 1 1 記載の情報処理方法。

【請求項 1 3】 イメージデータを用いた印刷時間として描画処理時間、転送時間、プリンタの処理時間を含み、コードデータを用いた印刷時間としてコマンド生成時間、サブクローズ時間を含むことを特徴とする請求項 1 1 記載の情報処理方法。

【請求項 1 4】 イメージデータを用いた印刷時間として描画処理時間、転送時間、中間データ生成時間、シップ時間を含み、コードデータを用いた印刷時間としてコマンド生成時間、転送時間、シップ準備時間、シップ時間を含むことを特徴とする請求項 1 1 記載の情報処理方法。

【請求項 1 5】 前記選択ステップは、論理ページ単位で選択することを特徴とする請求項 1 1 記載の情報処理方法。

【請求項 1 6】 前記選択ステップにより選択された変換手段により変換されたデータにより印刷を行なうとプリンタでサブクローズが発生するか否かを判定する判定ステップと、

前記判定ステップによりサブクローズが発生すると判別された場合、変換手段により変換されるデータの階調を低下させるステップとを有することを特徴とする

請求項 1 1 記載の情報処理方法。

【請求項 1 7】 印刷データを調べ、描画不正が起こるか否かを判断する判断ステップを有し、

前記選択ステップは、前記判断ステップにより描画不正が起こると判断された場合、前記第 1 変換手段を選択し、前記判断ステップにより描画不正が起きないと判断された場合、前記第 2 変換手段を選択することを特徴とする請求項 1 1 記載の情報処理方法。

【請求項 1 8】 印刷データをイメージデータに変換する第 3 変換手段を利用する情報処理方法であって、

印刷データを調べ、描画不正が起こるか否かを判断する判断ステップとを有し、前記判断ステップにより描画不正が起きないと判断された場合、コードデータで処理する時間がイメージデータで処理する時間より長いかな否かを判定する判定ステップとを有し、

前記選択ステップは、前記判定ステップにより、コードデータで処理する時間がイメージデータで処理する時間より長いと判定された場合、前記第 3 変換手段を選択し、前記判定ステップにより、コードデータで処理する時間がイメージデータで処理する時間より長くないと判定された場合、前記第 2 変換手段を選択することを特徴とする請求項 1 1 記載の情報処理方法。

【請求項 1 9】 前記第 1 変換手段は RGB で論理演算を行ない、前記第 3 変換手段は YMCK で論理演算を行なうことを特徴とする請求項 1 8 記載の情報処理方法。

【請求項 2 0】 印刷データをイメージデータに変換する第 1 変換手段と、印刷データをプリンタでイメージデータを生成するコードデータに変換する第 2 変換手段と、を利用する情報処理方法であって

前記第 1 変換手段および前記第 2 変換手段のいずれかを論理ページ単位で選択する選択ステップとを有することを特徴とする情報処理方法。

【請求項 2 1】 印刷データをイメージデータに変換する第 1 変換手段と、印刷データをプリンタでイメージデータを生成するコードデータに変換する第 2 変換手段とを利用するプログラムを記憶する記憶媒体であって、

前記第 1 変換手段により変換させたイメージデータを用いた印刷時間と前記第 2 変換手段により変換させたコードデータを用いた印刷時間とを予測する予測ステップと、

前記予測ステップによる予測結果に基づきイメージデータを用いた印刷時間の方が短いかな否かを判別する判別ステップと、

前記判別ステップによりイメージデータを用いた印刷時間の方が短いと判別された場合、前記第 1 変換手段を選択し、前記判別ステップによりイメージデータを用いた印刷時間の方が短くないと判別された場合、前記第 2 変換手段を選択する選択ステップとを有するプログラムを記憶することを特徴とする記憶媒体。

【請求項 2 2】 前記予測ステップは、ホストコンピュータ側の処理時間、転送時間、プリンタ側の処理時間に基づき予測することを特徴とする請求項 2 1 記載の記憶媒体。

【請求項 2 3】 イメージデータを用いた印刷時間として描画処理時間、転送時間、プリンタの処理時間を含み、コードデータを用いた印刷時間としてコマンド生成時間、サブクローズ時間を含むことを特徴とする請求項 2 1 記載の記憶媒体。

【請求項 2 4】 イメージデータを用いた印刷時間として描画処理時間、転送時間、中間データ生成時間、シップ時間を含み、コードデータを用いた印刷時間としてコマンド生成時間、転送時間、シップ準備時間、シップ時間を含むことを特徴とする請求項 2 1 記載の記憶媒体。

【請求項 2 5】 前記選択ステップは、論理ページ単位で選択することを特徴とする請求項 2 1 記載の記憶媒体。

【請求項 2 6】 前記選択ステップにより選択された変換手段により変換されたデータにより印刷を行なうとプリンタでサブクローズが発生するか否かを判定する判定ステップと、

前記判定ステップによりサブクローズが発生すると判別された場合、変換手段により変換されるデータの階調を低下させるステップとを有することを特徴とする請求項 2 1 記載の記憶媒体。

【請求項 2 7】 印刷データを調べ、描画不正が起こるか否かを判断する判

断ステップを有し、

前記選択ステップは、前記判断ステップにより描画不正が起こると判断された場合、前記第 1 変換手段を選択し、前記判断ステップにより描画不正が起きないと判断された場合、前記第 2 変換手段を選択することを特徴とする請求項 2 1 記載の記憶媒体。

【請求項 2 8】 印刷データをイメージデータに変換する第 3 変換手段を利用するプログラムを記憶する記憶媒体であって、

印刷データを調べ、描画不正が起こるか否かを判断する判断ステップとを有し、前記判断ステップにより描画不正が起きないと判断された場合、コードデータで処理する時間がイメージデータで処理する時間より長いかな否かを判定する判定ステップとを有し、

前記選択ステップは、前記判定ステップにより、コードデータで処理する時間がイメージデータで処理する時間より長いと判定された場合、前記第 3 変換手段を選択し、前記判定ステップにより、コードデータで処理する時間がイメージデータで処理する時間より長くないと判定された場合、前記第 2 変換手段を選択することを特徴とする請求項 2 1 記載の記憶媒体。

【請求項 2 9】 前記第 1 変換手段は RGB で論理演算を行ない、前記第 3 変換手段は YMCK で論理演算を行なうことを特徴とする請求項 2 8 記載の記憶媒体。

【請求項 3 0】 印刷データをイメージデータに変換する第 1 変換手段と、印刷データをプリンタでイメージデータを生成するコードデータに変換する第 2 変換手段と、を利用するプログラムを記憶する記憶媒体であって

前記第 1 変換手段および前記第 2 変換手段のいずれかを論理ページ単位で選択する選択ステップを有するプログラムを記憶することを特徴とする記憶媒体。

【請求項 3 1】 オートモード、イメージモード、PDL モードのいずれかを設定させるための設定画面を表示させる手段と、

前記設定画面で設定されたモードに応じて、レンダラを選択し、選択されたレンダラを用いて印刷データを変換し、プリンタに転送する転送手段とを有することを特徴とする情報処理装置。

【請求項 3 2】 オートモード、イメージモード、PDL モードのいずれか

を設定させるための設定画面を表示させ、
前記設定画面で設定されたモードに応じて、レンダラを選択し、
選択されたレンダラを用いて印刷データを変換し、
変換したデータをプリンタに転送することを特徴とする情報処理方法。

【請求項 3 3】 オートモード、イメージモード、PDLモードのいずれかを
を設定させるための設定画面を表示させ、
前記設定画面で設定されたモードに応じて、レンダラを選択し、
選択されたレンダラを用いて印刷データを変換し、
変換されたデータをプリンタに転送するプログラムを記憶する記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、所定のアプリケーションプログラムからの印刷データを、印刷装置
が処理可能なデータ形式に変換処理し、印刷装置に出力する情報処理装置、情報
処理方法、及び、記憶媒体に関する。

【0002】

【従来の技術】

従来の印刷システムにおける印刷制御方法および処理の流れについて図 2 を用
いて説明する。

【0003】

一般的には、図 2 に示すようにセントロニクスインターフェースといったパラ
レル通信手段やネットワーク通信手段を介してホスト・コンピュータと印刷装置
とが接続され印刷システムを構成している。

【0004】

ホスト・コンピュータ側では、ワードプロセッサや表計算のようなアプリケー
ションソフトウェア 2 0 1（以下アプリと略称）がWindows（米国Microsoft社の
登録商標）のようないわゆる基本ソフトの上で動作している。このアプリにお
いて印刷を行う場合は、そのプログラム内から、基本ソフトが提供するいくつか
のサブシステムのうちグラフィック・サブ・システム 2 0 2 の機能を用いて行う

。このグラフィック・サブ・システムは、例えばWindowsでは、G D I (2 0 2 1) (Graphic Device Interface)と呼ばれておりディスプレイやプリンタに対する画像情報の処理を司っている。このG D I (2 0 2 1) は、ディスプレイやプリンタといった各デバイス毎の依存性を吸収するためにデバイスドライバと呼ばれるモジュールを動的にリンクし、それぞれのデバイスに対する出力処理を行う。プリンタに対するこのモジュールはプリンタ・ドライバ2 0 2 2 と呼ばれる。このプリンタ・ドライバ2 0 2 2 では、その能力や機能などに応じてあらかじめデバイスドライバに実装することが決められているD D I (Device Driver Interface)と呼ばれる関数群を用意する必要がある。アプリのA P I (Application Programming Interface)コールをG D I (2 0 2 1) がデバイスドライバ用にデータ変換を行い、このD D I 関数群が適宜G D I (2 0 2 1) からコールされ所定の印刷処理が実行されるような仕組みになっている。G D I (2 0 2 1) では、このようにプリンタ・ドライバを介してアプリからの印刷要求をシーケンシャルに処理している。

【 0 0 0 5 】

このプリンタドライバの処理系は、P D L モードタイプ (2 0 2 3) とイメージモードタイプ (2 0 2 4 ~ 2 0 2 6) の 2 種類大別できる。

【 0 0 0 6 】

P D L モードタイプとは、印刷装置側にP D L (Printer Description Language)と呼ばれる文字コード、図形コマンド、制御コマンド等からなるコードデータを処理し、イメージデータを生成可能なコントローラを搭載するもの向けで、プリンタドライバはシステムからD D I を介して渡されるデータをP D L のコマンドに変換し次第、G D I (2 0 2 8) を介してプリンタに送出する。

【 0 0 0 7 】

プリンタドライバの処理はD D I で受け取ったデータからコマンドを生成するだけなので、ドライバにとっては軽い処理といえる。P D L モードタイプの特徴として、プリンタのPDLコマンドを解析する能力が高ければG D I から抽象度の高いデータを受け取り、そのまま送出できるので通常のページ印刷においては転送するデータサイズを小さくすることが可能である。

【 0 0 0 8 】

一方、イメージモードタイプでは、印刷装置側では高度な描画処理は行わないよう、プリンタドライバ側で確保したメモリ空間上に印刷イメージの展開を行い、それを印刷装置で直接印刷できる形態に変換し、印刷装置に送り印刷するものである。

【 0 0 0 9 】

PDLモードタイプ同様に、GDIからDDIインターフェースを介して渡されたデータを使ってイメージモード用プリンタドライバ（2024～2026）がバンドメモリ2027に描画処理を行い、1バンド分の描画処理が終了すると、GDI（2028）を介して印刷装置にビットマップデータを送出する。

【 0 0 1 0 】

どちらの方式を取った場合も、プリンタまたはホストマシンでバンドメモリまたはフルページ分のメモリ空間上に描画処理を行う必要がある。

【 0 0 1 1 】

しかし、カラープリンタではRGB各色8ビットで600DPI,A4フルページ分のメモリを確保しようとするすると96Mbyte必要であり、現状ではホストマシン、プリンタどちらでも確保することは困難である。そこで、カラープリンタの描画処理はバンドメモリを使う方法が提案されている。

【 0 0 1 2 】

また、プリンタエンジンに最終的に渡すデータ形式はRGBといった輝度値ではなく、YMCK値という印刷装置のトナー色を示す濃度値を各色1ビットまたは2ビットであるので、イメージモード、PDLモード共にメモリ削減の為にYMCK各色1ビットor2ビットで処理を行うのが一般的である。イメージモードではこのYMCK各色1ビットor2ビットデータを送ることで転送データサイズを小さく抑さえ、印刷装置側の負荷を軽くするものがある。

【 0 0 1 3 】

例えば、多角形図形の印字処理については、PDLタイプでは、印刷装置に折れ曲がる点座標と線幅、線の色情報、内塗り色情報等がコマンドとして転送され、プリンタ内部ではそのデータを解析し、内部のYMCK各色1ビットまたは2ビット

の1バンド分のメモリ空間を用いて出力形態のイメージデータに展開処理を行う。PDL方式は1ページに無限に印刷データが発生する可能性があり、近年のアプリケーションが出力する印刷データは増加の一途をたどっている。ところが、バンディング処理で印刷を行う場合1ページ分の描画直前の中間データを保持する必要があるため、省メモリプリンタでは処理速度低下の要因になっている。

【0014】

イメージモード方式では、こういった問題を回避するために存在する。

【0015】

印刷装置に比べてメモリ構成に余裕のあるホストマシン上で動くプリンタドライバ側でバンド単位もしくはフルページの描画メモリを持ち、そこに多角形を展開する。1バンド分もしくはフルページ分の描画処理が終了すると、印刷装置に対してイメージデータとして描画メモリの内容が転送され所定の画像を印刷し、出力する。

【0016】

前述したように、PDLモード、イメージモード共に省メモリのために、YMCK各色1, 2ビットというデバイス依存の色空間上で描画処理を行っているが、GDIが規定している論理演算描画処理は輝度空間上での処理であるため、描画不正が発生することがある。これを回避するため、YMCK値からrgb値に戻し（逆UCR(Under Color Remove)処理）論理演算処理することで対応しているが、完全な描画処理を保証しないばかりか、処理速度低下の原因になっている。

【0017】

そこで、ユーザは、通常はPDLモードで描画処理を行い、出力に時間が掛かりすぎる場合は、高速イメージモード（YMCK各色1ビットイメージモード）に切り替えて出力しなおす、不正出力結果が出た場合は、RGB24BPPイメージモードで出力しなおす。といった2度、3度手間をユーザにかけることに成っている。

【0018】

つまり、現状はこういった印刷モードの選択がユーザに任されており、全てのモードで印刷してみないと最適なモードが解らない。また、複数ページ印刷においてはタイプの異なるページが含まれていると、1モードでは出力時間が掛かる

、または不正出力ページが混じる。といった問題も発生する。

【 0 0 1 9 】

以上、各モードの特徴を図 4 にまとめた。

【 0 0 2 0 】

こういった問題を解決するために、各々の方式の得意なデータだけが流れるように自動的に切り替える方式が求められている。

【 0 0 2 1 】

【発明が解決しようとする課題】

- ① 抽象度の高い印刷データ形式（PDLモード）では転送データサイズ、プリンタ内部で保持するデータサイズが大きくなることがあり、プリンタ処理速度が落ちることがある。
- ② 抽象度の高い印刷データ形式（PDLモード）では複雑な図形が集中していると、描画速度がプリンタの排紙速度に間に合わないためバンディング処理不可能となり、特殊パスに入ることによって処理速度が落ちることがある。
- ③ 省メモリのために、YMCK各色1,2bitのメモリ空間で描画処理を行う処理系（通常のPDLモード、高速イメージモード）では、輝度空間を想定した論理演算処理が不正になる場合がある。
- ④ 抽象度の低い印刷データ形式（イメージモード系処理）は簡単な文字または図形のためのページでも描画面積に比例したイメージデータがプリンタに転送されるため、エンジンの最大スループットがでない。
- ⑤ 複数ページ印刷時に、全てのページに最適なモードで印刷することが困難である。

【 0 0 2 2 】

前述の課題のうち①～④は各処理系の特徴であり、欠点とは言い難い。本特許は、各処理系の長所で他の処理系の短所を補うことでシステム全体としての改善を目指すものである。

【 0 0 2 3 】

つまり、印刷データに応じて最適な変換手段を選択し、高速に印刷を行うことを目的とする。

【 0 0 2 4 】

また、N - u p 印刷を高速に行うことを目的とする。

【 0 0 2 5 】

【課題を解決するための手段】

上記目的を解決するため本発明の情報処理装置は、
印刷データをイメージデータに変換する第 1 変換手段と、
印刷データをプリンタでイメージデータを生成するコードデータに変換する第 2 変換手段と、
前記第 1 変換手段により変換させたイメージデータを用いた印刷時間と前記第 2 変換手段により変換させたコードデータを用いた印刷時間とを予測する予測手段と、
前記予測手段による予測結果に基づきイメージデータを用いた印刷時間の方が短いかな否かを判別する判別手段と、
前記判別手段によりイメージデータを用いた印刷時間の方が短いと判別された場合、前記第 1 変換手段を選択し、前記判別手段によりイメージデータを用いた印刷時間の方が短くないと判別された場合、前記第 2 変換手段を選択する選択手段とを有することを特徴とする。

【 0 0 2 6 】

また、本発明の情報処理装置は、
印刷データをイメージデータに変換する第 1 変換手段と、
印刷データをプリンタでイメージデータを生成するコードデータに変換する第 2 変換手段と、
前記第 1 変換手段および前記第 2 変換手段のいずれかを論理ページ単位で選択する選択手段とを有することを特徴とする。

【 0 0 2 7 】

【発明の実施の形態】

本実施の形態の構成を説明する前に、本実施の形態を適用する印刷システムの印刷を担うカラーレーザービームプリンタ（以下「カラー L B P」と記述）の構成について図 1 を参照しながら説明する。

【 0 0 2 8 】

図 1 は、6 0 0 ドット／インチ（d p i）の解像度を有し、各色成分各画素が 8 ビットで表現された多値データに基づいて画像記録を行うカラー L B P の構造を示す側断面図である。図 1 において、1 0 0 はカラー L B P 本体であり、外部に接続されているホスト・コンピュータなどから供給されるプリントデータ（文字コードや画像データ等）及び制御コードから成る印刷データを入力して記憶するとともに、それらの情報に従って対応する文字パターンやイメージ等を作成し記憶媒体である記録紙上に像を形成する装置である。

【 0 0 2 9 】

1 1 0 はホスト・コンピュータから供給される印刷データを解析し印刷イメージの生成処理を行うとともにカラー L B P 本体 1 0 0 の制御を行うフォーマッタ制御部である。また、フォーマッタ制御部 1 1 0 は、ユーザによる操作およびユーザに対する状態通知のためのスイッチおよび L E D 表示器等が配されているオペレーション・パネル部 1 2 0 と接続されており、そのパネル部は印刷装置 1 0 0 の外装の一部として配設されている。フォーマッタ制御部 1 1 0 において生成された最終的な印刷イメージはビデオ信号として出力制御部 1 3 0 に送出され、出力制御部 1 3 0 は印刷装置 1 0 0 の不図示の各種センサからの状態入力とともに光学ユニット 1 4 0 および各種駆動系機構部に対し制御信号を出力し印刷装置 1 0 0 としての印刷処理を制御を司るものである。

【 0 0 3 0 】

図 1 に示す印刷装置において、給紙カセット 1 6 1 から給紙された用紙 P はその先端をグリッパ 1 5 4 f により挟持されて、転写ドラム 1 5 4 の外周に保持される。光学ユニット 1 4 0 により感光ドラム 1 5 1 上に形成された各色の潜像は、イエロー（Y）、マゼンタ（M）、シアン（C）、ブラック（B）の各色現像器 D y、D m、D c、D b により現像化されて、転写ドラム外周の用紙に複数回転写されて多色画像が形成される。その後、用紙 P は転写ドラム 1 5 4 より分離されて、定着ユニット 1 5 5 で定着され、排紙部 1 5 9 より排紙トレイ部 1 6 0 に排出される。ここで各色の現像器 D y、D m、D c、D b は、その両端に回転支軸を有し、各々がその軸を中心に回転可能となるように現像器選択機構部 1 5

2に保持される。これによって、各現像器D_y、D_c、D_b、D_nは、図1に示すように現像器選択のために現像器選択機構部152が回転軸152aを中心に回転しても、その姿勢を一定に維持できる構成をとっている。選択された現像器が現像位置に移動後、現像器選択機構部152は現像器と一体で支点153bを中心にして、選択機構保持フレーム153がソレノイド153aにより感光ドラム151方向へ引っ張られ、感光ドラム151方向へ移動し現像処理が行われるように構成されている。次に、帯電器156によって感光ドラム151が所定の極性に均一に帯電される。フォーマッタ制御部110において画像イメージとして展開された印刷情報は、対応するパターンのビデオ信号に変換されレーザードライバに出力され半導体レーザー141を駆動する。入力されたビデオ信号に応じて半導体レーザー141から発射されるレーザー光はオンオフ制御され、さらにスキナモータ143によって高速回転するポリゴンミラー142で左右方向に振らされ、ポリゴンレンズ134、反射鏡144を介して感光ドラム151上を走査露光する。これにより、感光ドラム151上には画像パターンの静電潜像が形成されることになる。次に、例えば、M（マゼンタ）色の静電潜像がM（マゼンタ）色の現像器D_mにより現像され、感光体ドラム151上にM（マゼンタ）色の第1のトナー像が形成される。一方、所定のタイミングで転写紙Pが給紙され、トナーと反対極性（例えばプラス極性）の転写バイアス電圧が転写ドラム154に印加され、感光体ドラム151上の第1トナー像が転写紙Pに転写されると共に、転写紙Pが転写ドラム154の表面に静電吸着される。その後、感光ドラム151はクリーナー157によって残留するM（マゼンタ）色トナーが除去され、次の色の潜像形成及び現像行程に備える。以下同様の手順によってC（シア）、Y（イエロ）、B_k（ブラック）の順で第2、3、4色目のトナー像の転写が行われる。但し、各色の転写時には、転写ドラム154には前回よりも高いバイアス電圧が印加される点は異なる。4色のトナー像が重畳転写された転写紙Pの先端部が分離位置に近づくと、分離爪158が接近してその先端が転写ドラム154の表面に接触し、転写紙Pを転写ドラム154から分離させる。分離された転写紙Pは定着ユニット155に搬送され、ここで転写紙上のトナー像が定着されて排紙トレイ160上に排出される。本実施の形態のカラーレーザービーム

プリンタは、以上のような画像形成過程を経て600ドット／インチ（dpi）の解像度で画像出力を行う。なお、本発明を適用可能なプリンタは、カラーLB Pに限られるものではなく、インクジェットプリンタやサーマルプリンタ等、他のプリント方式のカラープリンタでもよい。

【0031】

次に、印刷装置100におけるフォーマッタ制御部110について図2を用いて説明する。このフォーマット制御部は、通常はPDLコントローラなどとも呼ばれている部分であり、ホスト・コンピュータとの接続手段であるところのインタフェース（I/F）部111と、受信データ等を一時的に保持管理するための受信バッファ1121、送信データ等を一時的に保持管理するための送信バッファ1122、印刷データの解析を司るコマンド解析部113、印刷制御処理実行部114、描画処理実行部115、ページメモリ116等より構成されている。

【0032】

インタフェース（I/F）部111は、ホスト・コンピュータ200との印刷データの送受信を行う通信手段あり、通信プロトコルとしてIEEE-1284に準拠した通信を可能とするものである。但し本発明では、この通信手段に限定するものでなく、ネットワークを介してさまざまなプロトコルによる接続であってもよいし、IEEE-1394に準じた通信手段であってもよい。このインタフェース部111を通して受信した印刷データは、そのデータを一時的に保持する記憶手段である受信バッファ1121に逐次蓄積され、必要に応じてコマンド解析部113または描画処理実行部115によって読み出され処理される。コマンド解析部113は、各PDLコマンド体系や印刷ジョブ制御言語に準じた制御プログラムにより構成されており、文字印字、図形、イメージなどの描画に関する印刷データの解析結果は、描画処理実行部115に指示を与えて処理し、給紙選択やリセット命令などの描画以外のコマンドは、印刷制御処理実行部114に指示を出し処理する。

【0033】

描画処理実行部115では、文字やイメージの各描画オブジェクトをページメモリ116に逐次展開して行くレンダラである。図1で前述したカラーLB Pに

対しては、M C Y K の面順次でデバイス依存ビットマップデータを送出する必要があるが、標準状態では、そのために必要なメモリをすべて確保するわけではなく、1プレーン(1,2 or 4bit/pixel)の数分の1のバンド領域としてメモリが確保され、そのバンド領域を使いまわして画像をエンジン速度に同期して処理するように構成されている。通常はこのようにY M C K レンダラによる展開処理とプリンタエンジンへのビデオ信号の SHIPPING の追いかけっこ、つまりバンディング制御によってページメモリ 1 1 6 は管理されているが、十分なメモリがある場合は1ページ分が展開可能な領域を確保しても良い。なお、一般的に、フォーマッタ制御部 1 1 0 は、中央演算処理装置 (C P U) 、リードオンリーメモリ (R O M) 、ランダムアクセスメモリ (R A M) などを用いたコンピュータ・システムによって構成されている。また、各部の処理は、マルチタスクモニタ (リアルタイム O S) のもとでタイムシェアリングに処理される構成であっても良いし、各機能ごとに専用のコントローラ・ハードウェアを用意して独立して処理される構成であってもかまわない。

【 0 0 3 4 】

オペレーション・パネル 1 2 0 は、前述した通り印刷装置の各種状態を設定・表示するためのものである。出力制御部 1 3 0 は、ページメモリ 1 1 6 の内容をビデオ信号に変換処理し、プリンタ・エンジン部 1 5 0 へ画像転送を行う。プリンタ・エンジン部 1 5 0 は受け取ったビデオ信号を記録紙に永久可視画像形成するための印刷機構部であり、図 1 において前述したものである。以上、印刷装置 1 0 0 について説明したが、次にホスト・コンピュータ 2 0 0 を含む本実施例の印刷システムの全体構成について説明を加える。図 2 において 2 0 0 はホスト・コンピュータであり、プリントデータ及び制御コードから成る印刷情報を印刷装置 1 0 0 に出力するものである。ホスト・コンピュータ 3 0 0 は、入力デバイスであるところのキーボード 3 1 0 やポインティングデバイスであるところのマウス 3 1 1 と、表示デバイスであるディスプレイ・モニタ 3 2 0 を合わせた一つのコンピュータ・システムとして構成されている。ホスト・コンピュータ 3 0 0 は、Windows NT などの基本 O S によって動作しているものとする。

【 0 0 3 5 】

ホスト・コンピュータ側について、本発明に関する機能的な部分にのみ注目し、基本OS上での機能を大きく分類すると、アプリケーションソフトウェア301、グラフィック・サブ・システム302、印刷情報格納手段および印刷装置との通信手段を含むスプール・サブ・システム303に大別される。

【0036】

アプリケーションソフトウェア301は、例えば、ワープロや表計算などの基本ソフトウェア上で動作する応用ソフトウェアを指すものである。グラフィック・サブ・システム302は、基本OSの機能の一部であるGraphic Device Interface（以後、GDIと記す）3021とそのGDIから動的にリンクされるデバイスドライバであるところのプリンタ・ドライバ3022によって構成されている。

【0037】

プリンタドライバ3022は、GDIからDDI(Device Driver Interface)というインターフェースを介してコールされ、デバイスに応じた処理を描画オブジェクト毎に行うものである。本システムでは、DDI関数に渡された情報をページ単位でPDF3031（ページデータファイル）としてスプールするわけだが、その際に、プリンタドライバ3022から描画情報をチェックするモジュールであるPreCheckモジュール3023をコールすることで、そのページに含まれるデータ解析を行う。1ページ分のデータ解析することによって、そのページに含まれる描画命令を種類別にコール回数、データサイズの集計結果が生成される。またページ内の描画領域を最外接矩形として求まる。以上の結果はGDIを通じて対応するPDF3031と関連付けられた名前をつけられてPCF3032（実体はRAMまたはハードディスク上）（プレチェックファイル）としてセーブする。

【0038】

スプール・サブ・システム303は、グラフィック・サブ・システム302の後段に位置するプリンタ・デバイスに特有のサブ・システムでありデスプーラと呼ばれている。

【0039】

一旦、刷情報をスプールすることによって、デスプール時に出力デバイスを変

更したり、複数ページを扱ったページレイアウトを変更が可能になる。

【 0 0 4 0 】

さらに、本実施例においては、出力デバイスの描画命令ごとの処理速度や不正印字が発生するコマンド等が記載されたデバイス情報 3 0 4 1、ホストマシンの環境情報（空きディスク容量、通信速度）、スプール時に生成したページ単位のチェック結果である P C F 3 0 3 2 を読み込み、出力に適したトランスレータ（3 0 3 5 ～ 3 0 3 7）を、ページ単位、バンド単位、またはマトリクス単位に決定するものである。

【 0 0 4 1 】

印字するのに適したトランスレータ選択とは、モードの自動切換えポリシーによって変化する。例えば、高速印字優先ならば、高速に処理が出来るようなトランスレータを選択、画質優先ならば、不正印字が発生しないことを前提として可能な限り高速な処理系を選択することを示す。

【 0 0 4 2 】

選択されたトランスレータ 3 0 3 5 ～ 3 0 3 7 は、イメージモード系処理（3 0 3 4 ～ 3 0 3 7）であれば、バンドメモリ 3 0 3 8 を用いてイメージ展開を行いデバイス依存ビットマップ形式の PDL コマンドを生成する。PDL モード系処理（3 0 3 4）であればスプールファイルから PDL コマンドへの変換処理を行う。生成した PDL コマンドは GDI（3 0 3 9）、I / F 3 0 4 0）を介してプリンタにデータを送出される。

【 0 0 4 3 】

基本 OS によって、上述したこれらの名称や機能的な枠組みは若干異なる場合があるが、本発明で言う各技術的手段が実現できるモジュールであれば、それらの名称や枠組みは本発明にとってあまり大きな問題ではない。例えば、スプーラやスプールファイルと呼ばれるものは、別の OS においてプリント・キューと呼ばれるモジュールに処理を組み込むことによっても実現可能である。なお一般的に、これらの各機能モジュールを含むホスト・コンピュータ 3 0 0 は、中央演算処理装置（CPU）、リードオンリーメモリ（ROM）、ランダムアクセスメモリ（RAM）、ハードディスクドライブ（HDD）、各種入出力制御部（I / O

）などのハードウェアのもとで、基本ソフトと呼ばれるソフトウェアがその制御を司り、その基本ソフトの元で、それぞれの応用ソフト、サブ・システム・プロセスが機能モジュールとして動作するようになっている。

【 0 0 4 4 】

図 1 1 の模式図を使って、本実施例の理想的な実施形態を説明する。

【 0 0 4 5 】

(ホスト側プリンタドライバのさらに詳しい説明)

本発明を実現するためには、ホストマシンで動くプリンタドライバで最低限以下の機能が必要である。

- 1) 描画データスプール処理
- 2) チェック&ドライバ切替処理
- 3) 各種ドライバ

【 0 0 4 6 】

以下にそれぞれについての説明を行う。

【 0 0 4 7 】

- 1) 描画データスプール処理

WindowsNT4.0を例にとり、図 9 を用いて説明を行う。

【 0 0 4 8 】

WindowsNT4.0においては、GDIからコールされる描画用DDI関数として、以下のものが挙げられる。

DrvCopyBits(), DrvBitBlt(), DrvStretchBlt(), DrvStrokePath(), DrvFillPath()
DrvStrokeAndFillPath(), DrvPaint(), DrvTextOut()

これらの関数には引数を介して、GDIから描画オブジェクトが渡される。

【 0 0 4 9 】

例えば、イメージの描画処理関数であるDrvCopyBits()の引数は

```

BOOL  DrvCopyBits(
        SURF_OBJ    *psoDest,    //描画先メモリに繋がるハンドル
        SURF_OBJ    *psoSrc,     //ソースデータに繋がるハンドル
        CLIP_OBJ    *pco,        //クリップ情報のハンドル
    
```

```

XLATEOBJ  *pxlo,          //色変換情報のハンドル
RECTL     *prclDest,      //描画先矩形領域
POINTL    *pptlSrc        //ソースデータの左上の座標
)

```

となっている。

【 0 0 5 0 】

PDLモードでは、まず、pcoからクリップデータを獲得し、クリッピング指定データを発行し、その後、psoSrcと、pptlSrcが示す先のソースイメージデータを、psoDestと、prclDestが示す先に描画するようにイメージ描画データを生成して発行することになる。イメージモードでは、psoDestと、prclDestが示す先にpsoSrcとpptlSrcが示すイメージをpcoが示す領域に限りコピー処理を行えば良い。

【 0 0 5 1 】

本発明を実現するためには、1 ページ分の描画データを知った上で、最適な処理モードを決定し、そのモードで処理を実行する必要がある。そのためには、DDI 関数に渡されたデータを一旦スプールし、その際に最適なモードを決定。決定した後に、スプールしたデータを決定したモードで実行することで実現可能である。

【 0 0 5 2 】

そのためには、DDI 関数に渡されるデータの中でデスプール時にどのモードに決定しても処理可能となるデータをスプールする必要がある。

【 0 0 5 3 】

まず、スプールデータはページ単位で保持する必要がある。そこで、図 9 の P D F 9 0 0 は、ページを特定するためのpageID(9 0 1)に続いて、ページのサイズや、データサイズ、格納したオブジェクト数等を保持するページ情報(9 0 2)をヘッダとして持つ。

【 0 0 5 4 】

その後ろに続いて、DDI 関数でコールされた描画関数を示す番号9 0 3が入る。ここではDrvCopyBitsを例に説明を行う。

【 0 0 5 5 】

datasize(9 0 4)はDrvCopyBitsのプールにかかったサイズ、
soDset(9 0 5)はDrvCopyBits()関数に渡される描画先のサーフェースデータを
示すpsoDestで獲得可能なデータを格納。

【 0 0 5 6 】

SoSrc(9 0 6)はDrvCopyBits()関数に渡されるソースサーフェースデータを
示すpsoDestで獲得可能なデータを格納。

【 0 0 5 7 】

ソースイメージデータ(9 0 7)はpsoSrcから獲得可能なソースイメージデ
タの実態である。

【 0 0 5 8 】

co(9 0 8)はDrvCopyBits()関数に渡される描画先のクリップデータを示すpco
から獲得可能なデータを格納。また、クリップデータが存在する場合は、その後
ろにクリップデータ(9 0 9)を格納する。

【 0 0 5 9 】

xlo(9 0 1)はDrvCopyBits()関数に渡される色変換データを示すpxolで獲得
可能なデータを格納。

【 0 0 6 0 】

rclDest(9 1 1)はDrvCopyBits()関数に渡される描画先のサーフェース上の
矩形を示すprclDestから獲得可能なデータを格納。

【 0 0 6 1 】

ptlSrc(9 1 3)はDrvCopyBits()関数に渡されるソースサーフェース上のど
の位置から描画を行うかを示すpptlSrcから獲得可能なデータを格納。

【 0 0 6 2 】

以降、描画関数毎にそれぞれに必要なデータのプールが1ページ分続く。

【 0 0 6 3 】

デスプール時には、デスプーラが選択されたドライバのDDI関数にスプールし
た情報に戻して渡すことで、DDI関数仕様のままでこれまでのプリンタドライ
バが動作可能である。

【 0 0 6 4 】

2) のチェック&ドライバ切替処理

チェック&ドライバ切替手段は、プリンタドライバに渡される描画データをスプールする際に以下の点をチェックすることでスプールデータのデータを解析するものである。そこで、本実施の形態においては、(1)の描画データをスプールする際にモード切替にかかわるデータを獲得し、図10のPCF(プレチェックファイル)のようなデータ構造体にセットすることで実現するものである。

【 0 0 6 5 】

①カラーデータチェック

チェック&ドライバ切替処理は、カラープリンタであっても、モノクロモードをデフォルト状態とし、描画データの中にカラーデータがひとつでもある時には、カラーモードとして動作するようにする。PCF1000においては、1003には設定カラーモードを登録するが、desiColMode1004、各バンド情報のcolflg(10098)には初期化時にモノクロモードであることを示す値をセットする。描画データをスプールする際に、有彩色を指定するオブジェクトがあった場合は、desiColMode(1004)と、対応するバンドのcolflg(10098)をカラーモードにセットしなおす。

【 0 0 6 6 】

ページ全体のチェック終了後、ページのdesiColMode(1004)がモノクロモードであつたら、そのページはモノクロモードとして動かすことでドラムの回転数を1回に押さえる。ページのdesiColMode(1004)はカラーモードでも、colflg(10098)がモノクロモードで、かつDeviceイメージモードでの出力に決定されていた場合は、そのバンドは、K1(2, 4)BPPイメージモードでの描画処理を行うことで、同じバンドメモリサイズで大きなバンド領域の描画処理が可能となる。

【 0 0 6 7 】

カラーLBPプリンタは、カラーモードで印刷時は感光ドラム154が4回、回することでYMCKの色を乗せているが、モノクロモード時には、ブラクトナーのみを乗せるために1回、回るだけでり、モノクロデータしかないページであればカ

ラーモード時でもモノクロモードで動くことで高速化に繋がる。

【0068】

②データサイズチェック

プリンタがPDLモードで動くときに、「内部メモリに1ページ分の描画オブジェクトが入りきらない」時にはバンディング処理不可能となるため、処理速度が極端に落ちる。

【0069】

そこで、本実施の形態は、PDLモード時のプリンタ内部のデータサイズを予測することで、回避するものである。

【0070】

そこで、PDLモードで送ったときのプリンタ内部でのデータタイプとデータサイズをスプールデータから予測し、ページ毎にPDLDataSize(1006)に積算していく。また、本実施の形態では、モード切替をバンド単位で行うために、チェック用に設定した小さめのバンド高さでページを区切りそれぞれのデータを格納するデータ構造体をPCF内にバンド数分持っている。1009

【0071】

ページ毎のPDLDataSize(1006)を関連するバンド領域で割って、それぞれのバンドのPDLDataSize(10096)にセットする。

【0072】

一方、イメージモードで送った場合のデータサイズを求めるために、その描画データの描画領域の最大矩形サイズを、各バンド情報のBandRectLに更新する。各バンドで全バンド領域BandRectL(10094)をイメージデータで送った場合よりも、PDLDataSize(10096)が大きくなった場合は、そのバンドはDeviceイメージモード(YMCK、各色1, 2, 4ビット)で処理するように決定し、desiMode(10092)にその旨記録する。

【0073】

③描画速度チェック

PDLモードでバンディング処理によって省メモリを実現しているプリンタは、描画オブジェクトが同一バンド内に沢山あると、描画時間がシップ時間に間に合

わなくなり、バンディング処理による描画が不可能となる。通常プリンタではこういった場合、1ページ分のメモリサイズを獲得する特殊パスに入ることによって出力しているが、プリンタ内部のメモリに余裕がないと、バンドメモリ上に描かれたイメージデータの圧縮、伸張処理が頻発し処理速度低下につながる。

【0074】

そこで、本実施の形態では、単位面積当りの描画処理時間の予測を行い、シッブ時間に間に合わない領域はDeviceイメージモード（YMCK、各色1，2，4ビット）に展開して送ることに決定する。またはページ全体をDeviceイメージモードで送るといった方法でプリンタの処理を軽くすることが可能である。

【0075】

本実施の形態では、②と同様にPDLモードで送ったときのプリンタ内部でのデータタイプと描画処理予測時間を予測し、描画領域に対応するバンド毎にrendTime（10093）に積算していく。

【0076】

この値が、シッブ時間よりも大きくなったバンドはdesiModeをDeviceイメージモード（YMCK、各色1，2，4ビット）にセットする。

【0077】

④論理演算値チェック

プリンタでの描画処理は、ディスプレイの出力結果と同じになることを目的としているものである。そのため、ディスプレイで用いられているRGB空間（輝度色空間）での論理演算処理や、 α チャンネルを使った演算処理をプリンタドライバにそのまま指定してくるものが多々ある。ところが、プリンタはトナーやインクを使って印刷するために、濃度色空間値であるYMCK各色1または2ビットでバンドメモリまたはページメモリを確保し描画プレーンとして用いている。そのため、印刷結果が期待する論理演算結果にならないことがある。これを回避するためには、RGB各色8ビット以上のメモリ空間を使ったメモリ上で描画処理を行うRGB24BPPイメージモードまたは、プリンタ内部にRGBレンダリングパスを作る必要がある。

【0078】

本実施の形態では、こうしたケースではRGB 2 4 BPPイメージモードで処理するよう切りかえることで不正描画を回避するものである。

【 0 0 7 9 】

WindowsNT4.0においては、DrvBitBlt()の第10引数にrop4というパラメータがあり、この値でソースデータ、デスティネーションデータ、マスクデータ、ブラシデータの論理演算処理を指定してくる。チェックシステムで、この値をチェックし、濃度値で演算すると不正描画が発生する番号が来た時は、そのオブジェクトを描画するバンド領域のdesiMode (1 0 0 9 2)、またはページのdesiMode (1 0 0 5) をRGB 2 4 BPPイメージモードで処理することに決定する。

【 0 0 8 0 】

以上のチェックを描画オブジェクト毎に行うことによって、最適なモードを選択することが可能となる。モード切替の単位は最大ジョブ単位、最小オブジェクト単位が可能ではあるが、モードを切り替えることによる速度的効果を期待するならば、最低限ページ単位の切り替え処理が必要である。

【 0 0 8 1 】

次に、バンド単位のモード決定方法をフローチャートを用いて説明する。

【 0 0 8 2 】

ここでいうバンド単位とは、プリンタまたはドライバで描画処理に用いるバンドを示すものではなく、モード切替のチェック専用のバンドを示すものである。

【 0 0 8 3 】

バンド高さサイズは、プリンタドライバで行う描画処理の中で最も小さなバンド高さ以下であることが望ましい。

【 0 0 8 4 】

また、本実施の形態においては、プリンタ100は、プリンタドライバから1ページ内に複数種類のモード（イメージモード、PDLモード）の印刷データが送られた場合に処理可能でなくては成らない。

【 0 0 8 5 】

図5は描画データがプリンタドライバに渡されたときに行う処理の流れを示しており、描画データがコールされるたびに本処理が実行される。なお、図5の

フローチャートに係わるプログラムはホストコンピュータ300のメモリに記憶され、CPUによって実行される。

S501: 本オブジェクトにはカラーデータが入っているかチェック、カラーデータがある場合は、関係するバンドのカラーオブジェクトフラグをTRUEにセット。

S502: 本オブジェクトを描画する範囲がかかる全バンドはRGB24BPPイメージモードに決定しているか。そうならS509へ、それ以外のモードがあればS503へ、

S503: 本オブジェクトの描画論理はデバイス色空間上で可能か。可能であればS504へ、不可能ならS508へ、

S508: 関係するバンド領域をRGB24BPPイメージモードで処理することに決定。S509へ、

S504: 描画範囲がDeviceイメージモード(YMCK各色1, 2, 4ビットイメージモード)。YESならS509へ、それ以外(PDLモード)ならS505へ、

S505: 本オブジェクトをPDLモードで転送した場合のプリンタ内部で生成されるデータサイズを計算し、PDLモードで転送した場合のトータルデータサイズに加算。S506へ、

S506: 本オブジェクトをPDLモードで転送した場合に関係するバンドの描画時間を算出し、それまでのオブジェクトとの積算描画時間が排紙時間に間に合わない場合は、S507へ、間に合う場合は、S509へ、

S507: 間に合わないバンド領域をDeviceイメージモードに決定する。S507では、ユーザの設定に基づきYMCK1ビットモード、2ビットモード、及び、4ビットモードのいずれかにセットし、S509へ、

S509: イメージモードで送った場合、関係するバンドの描画領域を更新する。

【0086】

次に、1ページ分の全オブジェクトのプール処理と、チェック処理が終了したあと処理の説明を、図6のフローチャートを使って行う。なお、図6のフローチャートに係るプログラムはホストコンピュータ300のメモリに記憶され、C

PUによって実行される。

S601: 図7の700におけるチェック用バンドのカラーオブジェクトフラグが立っているかチェック。立っている場合はS602へ、立っていない場合は603へ、

S602: 本ページをカラーモードにセット。S603へ

S603: 本バンドはPDLモードか。YESならS604へ、それ以外ならS607へ、

S604: イメージモードで送った場合と、PDLモードで送った場合のプリンタ内部のデータサイズを比較。PDLモードで送ったほうが大きくなる場合はS606へ、イメージモードの方が大きいときはS607へ、

S606: 本バンドをDeviceイメージモードにセットする。S606では、ユーザの設定に基づきYMCK1ビットモード、2ビットモード、及び、4ビットモードのいずれかにセットし、S607へ、

S607: 本ページの全バンド終了。YESならS608へ、NOならS601へ、

S608: 本ページの全バンドの決定したモードをページIDと結びつけてプレチェックファイル(PCF)としてスプール

【0087】

3) 最後に、決定したモードに従って各種ドライバで行う描画処理について図7を用いて説明する。

【0088】

(2) で700のページをチェックすると、710のように判断されたとする。700、710のバンド区切りは、チェック用の仮想バンドである。

【0089】

701は高解像度のイメージデータであり、RGB各色8ビット値のままPDLコマンドに変換すると、それだけで数10メガバイトになってしまう。そこで、第二第三バンドはYMCK4BPP(各色1ビット)イメージモードで描画することに決定している。

【0090】

702は、YMCK描画処理では描画不正が発生する描画データであるため、対応するバンドはRGB24BPPイメージモードに決定している。それ以外のバンドはPDLモードで問題ないと判断されていることを710は示している。

【0091】

デスプーラ3033は、こうして渡された仮想バンドデータを元に、デスプーラするためのバンド情報の作成を行う。

【0092】

イメージモードで処理するためのバンドメモリを確保し、確保できたバンドメモリサイズから各イメージモードで使用するバンド高さを算出し、バンド高さを設定しなおす。7201、7203

(同じメモリサイズでもRGB24BPPイメージモードとYMCK4BPPイメージモードではバンド高さが異なる)

【0093】

PDLモードはプリンタ内部でバンディング処理を行うため、プリンタドライバでは連続したPDL領域をつなぐ。7202

【0094】

こうして再構成したバンドの状態を720に示す。

【0095】

デスプーラ3033は、この情報に従ってバンドリストを作成し、各バンド毎に対応したドライバをコールすることで描画処理、またはPDLデータへの変換処理を行う。

【0096】

以上の処理を図8のフローチャートを使ってさらに詳しく説明する。

【0097】

なお、図8のフローチャートに係るプログラムはホストコンピュータ300に記憶され、CPUによって実行される。

【0098】

以下の処理は1ページ分の描画データがスプーラされた後、そのページをデスプーラする際の処理を示すものである。

vH = 仮想バンドの高さ (図 7 の 7 1 0 で示すバンド高さ)

$maxH$ = 獲得できたバンド用メモリを各レンダラで使う場合に取りうる最大高さ (図 7 の 7 2 0 1 の上部分、7 2 0 3 で示すバンド高さ)

$nowBH$ = 現在計算中の処理バンド高さ

S 8 0 1 : 出力を行うページに対応する P C F (スプール時にチェックした結果を格納したファイル) をオープンする。S 8 0 2 へ、

S 8 0 2 : 現ページの描画処理に必要とする描画モードを検出。現在計算中のバンド高さを格納する $nowBH$ の初期化処理も行う。S 8 0 3 へ、

S 8 0 3 : P D L モードだけの場合は、S 8 1 5 へ、イメージモードがあった場合は、S 8 0 4 へ、

S 8 0 4 : イメージモードとして使用可能なバンドメモリを獲得、獲得したバンドメモリサイズと、ページ横幅サイズ、各モードの $bit/pixel$ 値から描画処理用バンド高さの最大値 $maxH$ を求める。S 8 0 5 へ、

S 8 0 5 : 次仮想バンド情報獲得、S 8 0 6 へ、

S 8 0 6 : 一つ前の仮想バンドと同じ処理モードか。同じなら S 8 0 8 へ、違うなら S 8 0 7 へ、

S 8 0 7 : $nowBH$ に 0 以外の値が入っていたら、 $nowBH$ 高さのバンドを描画位置、描画モードと共に登録、 $nowBH=0$ に初期化、 $nowBH$ に 0 が入っていたらスルーする。

S 8 0 8 へ、

S 8 0 8 : $maxH$ よりも仮想バンド高さ vH と $nowBH$ を合わせた値が大きい場合は S 8 0 9 へ、小さい場合は S 8 1 0 へ

S 8 0 9 : 仮想バンド空間を $(vH+nowBH)/maxH$ 個 (小数部分を切り捨てた整数個、例えば、 $256/100=2.56$ の 2 の部分) の $maxH$ 高さのバンドに分割して描画モードと共に登録、 $nowBH = (vH + nowBH) \% maxH$ (割り算のあまり部分、例えば、 $256/100=2$ あまり 56 の 56 の部分) をセット、S 8 1 1 へ、

S 8 1 0 : $nowBH = vH$, S 8 0 5 へ、

S 8 1 1 : 全仮想バンドの読みこみ終了したら S 8 1 2 へ、まだなら S 8 0 5 へ、

S 8 1 2 : $nowBH$ に 0 以外の値が入っていたら、 $nowBH$ 高さのバンドを描画位置、

描画モードと共に登録、nowBH=0に初期化、nowBHに0が入っていたらスルーする。
S 8 1 3 へ、

S 8 1 3 : S 8 0 7, S 8 0 9 で登録したバンドデータを元に、バンドリストを作成し、スプールされている描画データを、バンドバンドリストに繋ぐ。S 8 1 4 へ、

S 8 1 4 : バンドリストに登録された上位バンドから、描画処理、またはPDLデータへの変換処理を行い、生成されたデータをプリンタに送出する。

S 8 1 5 : スプールされている描画データを読みこみ、登録されている順にPDLコマンドに変換し、プリンタに送出する。

【 0 0 9 9 】

なおS 8 1 4 では、S 5 0 7 でYMCKモードがセットされ、ユーザによりYMCK各色1BPPイメージモードが指定されていた場合、YMCKレンダラ3036でRGB24BPPイメージからYMCK1BPPイメージに変換し、YMCK1BPPイメージで演算、描画処理を行い、プリンタに出力する。

【 0 1 0 0 】

また、S 5 0 7 でYMCKモードがセットされ、ユーザによりYMCK各色2BPPイメージモードがセットされていた場合、YMCKレンダラ3036で、RGB24BPPイメージからYMCK2BPPイメージに変換し、変換したYMCK2BPPイメージで演算、描画処理を行い、YMCK2BPPイメージをプリンタに出力する。また、S 5 0 7 でYMCKモードがセットされ、ユーザによりYMCK4BPPイメージモードがセットされていた場合、YMCKレンダラ3036で、RGB24BPPイメージからYMCK4BPPイメージに変換し、変換したYMCK4BPPイメージで演算、描画処理を行い、YMCK4BPPイメージをプリンタに出力する。また、S 5 0 8 でRGBモードがセットされた場合、RGBレンダラ3037で、RGB24BPPイメージモードで演算、描画処理を行い、ユーザにより指定されたYMCK各色1、2、4BPPイメージに変換しプリンタに出力する。

S 8 1 3 : スプールされている描画データをそのままロードし、順番にPDLデータに変換し、プリンタに送出する。

【 0 1 0 1 】

(プリンタ側のさらに詳しい説明)

本発明を実現するためには、プリンタは以下の機能が必要である。

① バンド単位、ページ単位の処理モード切替機能

PDLプリンタには、デフォルトパスであるPDLモードと別にイメージモードというデバイスに依存したイメージデータを受け取り、そのまま出力するモードを持っているものがある。しかし、イメージモードは大量のイメージデータを高速に処理するために特別なメモリ構成で処理するために、同一ジョブではPDLモードとイメージモードを切り換えることは困難である。

【 0 1 0 2 】

本発明を実現するためには、ページ単位、さらにはバンド単位でPDLモードとイメージモードを切り替える必要がある。

【 0 1 0 3 】

そこで、本実施の形態では、PDLモードのイメージ描画データのデータタイプにデバイス依存イメージデータを受け付ける口を作ること、この問題を解決している。

【 0 1 0 4 】

また、バンド単位で、描画処理を行うモードを切りかえるためには、バンド間に跨って描画されるオブジェクトがある場合に不連続線が出ないように、描画アルゴリズムも揃えねば成らないことは言うまでもない。

【 0 1 0 5 】

(第二の実施の形態)

次に図 1 6 を用いて第二の発明の実施の形態を説明する。

【 0 1 0 6 】

バンド単位でイメージモード、PDLモードを切り替えるとつなぎ目に起きる場合が考えられるため、論理ページ単位で切り替える例を説明する。

【 0 1 0 7 】

1 6 0 1 は一般的なドキュメント作成アプリケーションを使って作成した 4 ページのドキュメントを示している。1 ページ目から 3 ページ目までは文字だけデ

ータであり、プリンタにとっては高速な印字が可能なページである。ところが4ページ目に貼り付けられているイメージデータ1602は、高解像度のカラーイメージである。4ページを1ページに合わせた(4-up)をPDLモードで処理すると、高解像度イメージがそのままPDLコマンドとしてプリンタに送出され、プリンタの処理速度が大幅に低下する。

【0108】

これを回避するために本発明を使って処理系を自動的に切り替えるわけだが、N-up印刷時には、以下の二つの切り替え方法が考えられる。

【0109】

1) 物理ページ単位で切り替える

高解像度イメージが含まれる物理ページ全体をイメージモードで処理するためには、ページ全体をいくつかのバンド領域に分割し1603、バンド単位で描画処理を実行し、生成したデバイスビットマップデータは、ページ全体のサイズとなる。(600DPI, 出力階調数2, YMCK印刷, A4サイズの場合、非圧縮時のデータサイズは32Mbyteとなる。そのため、プリンタに搭載されているメモリサイズによっては、指定階調数で出力できない可能性が高い。)

2) 論理ページ単位で切り替える

1, 2, 3、ページはPDLモード系トランスレータで文字コードを利用した文字描画命令のみが生成される。4ページ目は、イメージモード系トランスレータが選択される。イメージモード系トランスレータは、縮小処理を行った時の領域のバンドメモリを用意し1605、そこに縮小したイメージを描画するため、バンド分割をする回数も、生成されるデバイスビットマップデータも(1)の1/4となる。その分搭載メモリサイズが少ない場合でもイメージモード系処理で高階調出力の可能性が高い。

【0110】

本発明は、(2)の方法をとるものである。以下に実現方式について詳しく説明する。

【0111】

(ホスト側プリンタドライバのさらに詳しい説明)

本発明を実現するためには、ホストマシンで動くプリンタドライバで最低限以下の機能が必要である。

【0112】

- 1) 描画データスプール処理
- 2) スプール時の描画命令毎にチェック処理、描画領域保持処理
- 3) デスプール前に、デバイス情報、出力環境情報、レイアウト情報、チェック結果を読み込む処理
- 4) チェック結果から再生手段と出力階調数を論理ページ単位で選択する処理
- 5) 選択した再生手段で論理ページ毎に選んだ処理系でデスプール処理を行い、物理ページ単位でページ排出を行う処理

【0113】

以下にそれぞれについての説明を行う。

【0114】

- 1) 描画データスプール処理

Windows NT4.0を例にとり、図9を用いて説明を行う。

【0115】

Windows NT4.0においては、GDIからコールされる描画用DDI関数として、以下のものが挙げられる。

【0116】

DrvCopyBits(), DrvBitBlt(), DrvStretchBlt(), DrvStrokePath(), DrvFillPath(),
DrvStrokeAndFillPath(), DrvPaint(),
DrvTextOut()

【0117】

これらの関数には引数を介して、GDIから描画オブジェクトが渡される。

【0118】

例えば、イメージの描画処理関数であるDrvCopyBits()の引数は
BOOL DrvCopyBits(
SURFOBJ *psoDest, //描画先メモリに

繋がるハンドル

SURFOBJ * p s o S r c, //ソースデータに繋

がるハンドル

CLIPOBJ * p c o, //クリップ情報のハンドル

XLATEOBJ * p x l o, //色変換情報のハンドル

RECTL * p r c l D e s t, //描画先矩形領

域

POINTL * p p t l S r c //ソースデータの

左上の座標)

となっている。

【0119】

PDLモードでは、まず、p c oからクリップ情報を獲得し、クリッピング指定命令を発行し、その後、p s o S r cと、p p t l S r cが示す先のソースイメージデータを、p s o D e s tと、p r c l D e s tが示す先に描画するようにイメージ描画命令を生成して発行することになる。イメージモードでは、p s o D e s tと、p r c l D e s tが示す先にp s o S r cとp p t l S r cが示すイメージをp c oが示す領域に限りコピー処理を行えば良い。

【0120】

本発明を実現するためには、1ページ分の描画情報を知った上で、最適な処理モードを決定し、その後デスプूल処理を実行する必要がある。そのためには、DDI関数に渡されるデータの中でデスプूल時にどのモードでも再生処理可能なデータ形式でスプूलしなくてはならない。

【0121】

まず、スプूलデータはページ単位で保持する必要がある。そこで、図9のPDF900は、ページを特定するためのp a g e I D (901)に続いて、ページのサイズや、データサイズ、格納したオブジェクト数等を保持するページ情報(902)をヘッダとして持つ。

【0122】

その後ろに続いて、DDI関数でコールされた描画関数を示す番号903が入

る。ここではDrvCopyBitsを例に説明を行う。

【0123】

datasize (904) はDrvCopyBitsのプールにかかったサイズ、sodset (905) はDrvCopyBits () 関数に渡される描画先のサーフェース情報を示すpsodestで獲得可能な情報を格納。

【0124】

Sosrc (906) はDrvCopyBits () 関数に渡されるソースサーフェース情報を示すpsodestで獲得可能な情報を格納。

【0125】

ソースイメージデータ (907) はpsosrcから獲得可能なソースイメージデータの実態である。

【0126】

co (908) はDrvCopyBits () 関数に渡される描画先のクリップ情報を示すpcoから獲得可能な情報を格納。また、クリップデータが存在する場合は、その後ろにクリップデータ (909) を格納する。

【0127】

xlo (901) はDrvCopyBits () 関数に渡される色変換情報を示すpxolで獲得可能な情報を格納。

【0128】

rcldest (911) はDrvCopyBits () 関数に渡される描画先のサーフェース上の矩形を示すprcldestから獲得可能な情報を格納。

【0129】

ptlsrc (913) はDrvCopyBits () 関数に渡されるソースサーフェース上のどの位置から描画を行うかを示すpptlsrcから獲得可能な情報を格納。

【0130】

以降、描画関数毎にそれぞれに必要なデータのプールが1ページ分続く。

【0131】

デスポール時には、デスポーラが選択されたドライバのDDI関数にプール

した情報に戻して渡すことで、DDI関数仕様のままでこれまでのプリンタドライバが動作可能である。

【0132】

2) スプール時の描画命令毎にチェック処理、描画領域保持処理

スプール時の描画命令毎のチェック手段、描画領域保持手段は、プリンタドライバに渡される描画データをスプールする際に処理速度に関係する以下の点をチェックすることでスプールデータの情報を解析するものである。そこで、本実施例においては、(1)の描画命令をスプールする際にモード切替にかかわる情報を獲得し、ページ単位で図15のPCF(プレチェックファイル)1000のようなデータ構造体に格納、スプールすることで実現するものである。

【0133】

PCF1000のデータ構造は、そのファイルが属するジョブナンバー1001、ページナンバー1002で始まる構造体で、この二つの値により、対応するページを特定可能である。

【0134】

その他は、ページをY軸方向に分割したバンドを単位としたbandinf構造体1030で構成されている。

【0135】

仮想領域は、VbandSize1004に示されたサイズのバンドが、pageSize1003の中に、bandCnt1005分敷きつめられたバンド領域を仮想的に定義したものである。

【0136】

Bandlnfoは、その仮想バンドの領域を矩形情報で保持するBandRectL1010で始まる構造体であるが、この構造体には、BandRectL1010の領域内に関連する描画処理情報を記録するための物である。

【0137】

詳細は以下のとおりである。

【0138】

バンド内に描かれる描画処理の最大概説矩形領域DrawRectL1011

、バンド内に描かれる描画領域の中の有彩色領域ColorRectL1012、YMCK印字では不正印字が発生する領域ROPRectL1013イメージ系関数による描画処理領域ImgRectL1014、グラフィックス系関数による描画領域GraphicRectL1014、文字系描画領域TextRectL1016。

【0139】

以上の各RectL構造体が示す領域をわかりやすく図示したものを図11に示す。

【0140】

1101は、DrawRectL1011が保持している各仮想バンド内の描画領域を最外郭矩形領域を示すものである。

【0141】

1102は、ColorRectL1012が保持している各仮想バンド内の有彩色で描画処理を行う領域の最外郭矩形領域を示すものである。

【0142】

1103は、ROPRectL1013が保持している各仮想バンド内の描画領域の最外郭矩形領域を示す物である。

【0143】

1104は、ImgRectL1014が保持している各仮想バンド内のイメージ系描画領域の最外郭矩形領域を示す物である。

【0144】

1105は、GraphicRectL1015が保持している各仮想バンド内のグラフィックス系描画領域の最外郭矩形領域を示す物である。

【0145】

1106は、CharRectL1016が保持している各仮想バンド内の文字系描画領域の最外郭矩形領域を示す物である。

【0146】

以上の領域は、描画データスプール時に、描画命令から推測した描画領域の最外郭矩形領域を求め、それとbandRectLの重なり部分を示すものである

【0147】

B a n d l n f o 構造体1030は、それ以外に、DDI関数毎の情報も1017～1020に収集する。

【0148】

Windows NT40のDDIを例にとって説明すると、ビットマップの拡張指定可能なDDI関数であるStretchBlt関数についての情報は、StretchBlt1019に保存する。

【0149】

1019は、StretchBlt関数のコール回数callcnt10191、1BPPイメージの個数1BPPCnt10192、1BPPイメージのトータルデータサイズ1BPPTotalSize10193といったこのバンドに関連する情報を保持可能である。

【0150】

その他のDDI関数、CopyBits(1017)、BitBlt(1018)、Polygon(1020)も同様にそれぞれの関数I/Fに渡される情報を保持するようなデータ構造体を持っている。

【0151】

3) デスプूल前に、デバイス情報、出力環境情報、レイアウト情報、プレチェック結果を読み込む処理

【0152】

デバイス情報：

スプूल時に収集した情報から処理モードを決定する上で、プリンタの、グラフィックスデータの処理速度、イメージデータの処理速度、論理演算能力が異なると、予測結果が異なることがある。これらの値を、プリンタごとに求め、ファイル化したものをインストールファイルとしてシステムに保持しておく。

【0153】

出力環境情報：

プリンタドライバが動作するホストマシンの能力(CPUスピード、ディスク

速度、ディスク空き容量)、プリンタとホスト間の転送速度を意味する。

【0154】

レイアウト情報：

pageSize1003を実際にはどのサイズで出力するかによって、イメージモードで処理した場合の転送データサイズが大きく変化する。

【0155】

また、N-up指定等の1ページ辺りに複数のページを配置する場合には、PDLモードで高速な処理が可能だったページが集まり、プリンタの負荷が増えることがある。

【0156】

スプール時のページを論理ページ、出力時に実際に描画処理を行うページを物理ページと定義し、物理ページ内に、論理ページが如何に配置されるかを示すのがレイアウト情報である。

【0157】

プレチェック結果：

スプーラでチェックした結果を格納したPCFを示す。

【0158】

物理ページ内にかかわる全論理ページのPCFを読み込み、演算対象とする。

【0159】

4) チェック結果から再生処理と出力階調数を論理ページ単位で選択する処理
以下①～③の判定を論理ページ単位に行い、結果を管理テーブルに保持する。

【0160】

① カラーモノクロ判定

論理ページ単位でカラーモノクロモードを切り替える場合：

PCFの各バンドのColorRectL領域（有彩色オブジェクトの描画処理のあった領域）の有無をチェック、一つでも存在すれば、そのページはカラーモードで処理を行う。

【0161】

バンド単位で処理モードを切り替える場合：

PCFの各バンドのColor Rect L領域（有彩色オブジェクトの描画処理のあった領域）の有無をチェック、存在するバンドはカラーモードで処理を行う。

【0162】

② 不正印字回避

論理ページ単位で処理モードを切り替える場合：

PCFの各バンドのROP Rect L領域（YMCK色空間では不正印字が発生する論理演算処理のあった領域）の有無をチェック、一つでも存在すれば、そのページはRGB 24 BPPイメージモードで処理を行う。

【0163】

バンド単位で処理モードを切り替える場合：

PCFの各バンドのROP Rect L領域（YMCK色空間では不正印字が発生する論理演算処理のあった領域）の有無をチェック、存在するバンドはRGB 24 BPPイメージモードで処理を行う。

【0164】

③ 処理速度予測

イメージモード、PDLモードの処理速度は図6に示したような関係にある。

【0165】

「イメージモード処理速度予測」

イメージモード系処理600は、大きく分けると、プリンタドライバの処理時間と、プリンタの処理に分けられる。

【0166】

プリンタドライバの処理は、バンドメモリを使った描画処理時間1201であるが、色変換処理、2値化処理まで含んだ描画処理を行うため、PDLモードでのドライバの処理に比べて重い処理と言える。描画オブジェクト毎の描画処理情報に、ホストマシンのCPU速度、空きディスク容量を掛け合わせて処理時間の予測値を求める。

【0167】

プリンタ処理時間は、データ転送時間1202とプリンタ内部の中間データ生

成時間 1 2 0 3、シッ プ時間 1 2 0 4 を含んでいる。データ転送時間 1 2 0 2 を求めるために、まず描画面積を求める。

【 0 1 6 8 】

描画面積は図 1 5 で説明した各仮想バンドの Draw Rect L 領域とレイアウト情報を掛け合わせることで算出する。求めた描画面積に指定出力階調数をかけて非圧縮時のデータサイズが求まる。転送データサイズは、データ圧縮を行う場合、圧縮率が解らないと正確な値はわからないので、平均値を適用する。また、オブジェクト毎の描画領域はわかっているので、オブジェクト毎の圧縮率平均値を求めることで精度を向上させることは可能である。

【 0 1 6 9 】

プリンタ内部の中間データ作成時間 1 2 0 3 は、既にデバイスに依存したデータ形式にまで変換されているため、中間データの格納処理だけであり、処理時間は描画面積に比例しており、比例係数は機種依存値から取得する。

【 0 1 7 0 】

シッ プ時間 1 2 0 4 は、エンジンスループット速度に依存。

【 0 1 7 1 】

以上の合計値を求めることでイメージモードの処理時間が求まる。

【 0 1 7 2 】

「PDL モード処理速度予測」

PDL 系処理 1 2 0 5 も、イメージモード同様プリンタドライバの処理時間と、プリンタの処理に分けられる。

【 0 1 7 3 】

まず、プリンタドライバの処理は、基本的にはシステムから渡された描画命令をそのまま PDL コマンドに変換するコマンド生成処理 1 2 0 6 だけなので、イメージモード系ドライバの処理に比べて高速である。そのため、ホストマシンの CPU 速度、空きディスク容量の影響は受けにくい。

【 0 1 7 4 】

各オブジェクト毎のコマンド生成予測係数を元に予測する。

【 0 1 7 5 】

プリンタの処理時間に含まれるデータ転送時間 1 2 0 7 は P D L コマンドサイズと、転送経路の速度に影響を受ける。P D L コマンドのサイズは、出力モード（カラー、モノクロ等）、機種、言語のバージョンによって異なるので、デスプーラで収集した機種情報を使ってオブジェクト毎の P D L コマンドサイズを算出する。また、レイアウト処理により縮小する場合もデータサイズは変化するものがあるので、レイアウト情報も加味して算出する。

【 0 1 7 6 】

プリンタのエンジンスループットに間に合うように、受信したデータを加工する時間をシッパ準備時間 1 2 0 8 とした。プリンタのメモリ容量に中間データが収まる場合は、二つのバンドメモリを使って描画処理とシッパ処理を交互に行う、いわゆるバンディング処理が可能であり、この場合プリンタの処理速度はエンジンスループットで連続ページ印刷が可能である。ところが、プリンタの中間データ格納許容量を超えるデータが送られる、または、複雑な演算が必要でバンディング処理が間に合わない場合に、プリンタはバンディング処理を諦め、1 ページ分のメモリを確保する特殊処理モードに入る。この状態に入るとプリンタの処理速度は極端に落ちる。この時の時間予測は、搭載メモリサイズ、プリンタの C P U 速度、プリンタのハードレンダラー能力に依存する。

【 0 1 7 7 】

シッパ時間 1 2 0 9 は、エンジンスループット依存。

【 0 1 7 8 】

5) 選択した再生手段で、指定階調出力が可能かどうかを検証し、可能と判定した時には、出力階調数を落として再度再生手段を選択しなおす手段

【 0 1 7 9 】

高階調出力可能なプリンタでも、搭載メモリサイズが少ない時は、指定された階調数、解像度で出力できない場合も依存する。こういった場合でも、ページプリンタは P D L モード時には解像度、または階調数をプリンタ内部で自動的に低下（デグレード）させることで、出力不可能なことが無いようにしている。

【 0 1 8 0 】

ところが、イメージモード系処理で生成される印刷データの、デバイス依存ビ

ットマップ（既に出力階調数、解像度のハーフトニング処理を加えたデータ）はプリンタでデグレード処理を行うと画質低下が激しいため、プリンタのメモリ等裁量が出力が保証されない時は高階調指定が出来ないようになっている。

【 0 1 8 1 】

本実施例では、PDLモードとイメージモードを動的に切り替えるシステムを提供するために、搭載メモリサイズが少ない時でもPDLモード時のように高階調指定を可能とするものである。

【 0 1 8 2 】

これを実現するために、以下の方法を取っている。

【 0 1 8 3 】

1) 選択した処理系にイメージモードが含まれた場合は、物理ページ内に含まれる論理ページ全ての描画コマンドをプリンタ内部で中間データに変換した時のサイズ (Data Size) を予測

2) プリンタのメモリ搭載サイズを取得し、中間データ格納部のサイズ (Work Size) を算出

3) Data Size > Work Size の時には、出力階調数を落として再度、再生手段を選択しなおす。

【 0 1 8 4 】

以上の予測結果を組み合わせることで、最適な処理モードを選択する。

【 0 1 8 5 】

プリンタドライバ内での物理ページ内における論理ページ判断の流れを図 1 3、図 1 4 のフローチャートで説明する。なお、図 1 3、1 4 のフローチャートに係わるプログラムはホストコンピュータのメモリに記憶されており、CPUによって実行される。

【 0 1 8 6 】

「物理ページの処理系決定の流れ」

S 1 3 0 1 : 物理ページ内の論理ページ毎に最適な処理モード決定、S 1 3 0 2 へ進む。

S 1 3 0 2 : デバイス依存ビットマップが出力コマンド内にある場合は S 1 3 0

3 へ、ない場合は処理終了。

S 1 3 0 3 : 計算中の論理ページ内の印刷命令をプリンタ内部で中間データに変換した時のサイズを予測、それまでに算出したページの中間データサイズのトータル値 (t o t a l S i z e) に加算し、S 1 3 0 4 へ進む。

S 1 3 0 4 : プリンタに搭載されているメモリサイズから算出した中間データ格納領域サイズよりも、S 1 3 0 3 で求めたデータサイズのほうが大きい時は S 1 3 0 5 へ、小さい時は処理終了。

S 1 3 0 5 : 指定階調数よりも小さな階調数を設定。S 1 3 0 1 へ戻る。

【 0 1 8 7 】

「論理ページ毎の処理系決定の流れ」

S 1 4 0 1 : 論理ページ情報取得し、S 1 4 0 2 へ進む。

S 1 4 0 2 : P D L モードで不正印字発生が予測される場合は S 1 4 1 2 へ、無い場合は S 1 4 0 3 へ進む。

S 1 4 0 3 : P D L モードで処理した時のプリンタ内部で保持する中間データの種類、個数、サイズ予測し、1 4 0 4 へ進む。

S 1 4 0 4 : バンドレンダリングで処理不可能と判断した場合は、S 1 4 0 5 へ、バンドレンダリング可能なら S 1 4 0 6 へ進む。

S 1 4 0 5 : サブクローズによって発生する遅延時間を算出し、P D L モード処理時間に加算、S 1 4 0 6 へ進む。

S 1 4 0 6 : イメージモードで処理した時のプリンタ内部で生成される中間データサイズ、処理時間 (I M G T i m e) 予測、S 1 4 0 7 へ進む。

S 1 4 0 7 : P D L モードで処理した時のプリンタ内部のデータサイズ、処理時間 (P D L T i m e) 予測、S 1 4 0 8 へ進む。

S 1 4 0 8 : $I M G T i m e < P D L T i m e$ なら S 1 4 1 0 へ、そうでなければ S 1 4 0 9 へ進む。

S 1 4 0 9 : 処理中の論理ページを P D L モードに決定し、結果をテーブルに登録し、S 1 4 1 1 へ進む。

S 1 4 1 0 : 処理中の論理ページを Y M C K イメージモードに決定し、結果をテーブルに登録し、S 1 4 1 1 へ進む。

S1412：処理中の論理ページをRGBイメージモードに決定し、結果をテーブルに登録し、S1411へ進む。

S1411：論理ページがまだある場合はS1401へ、無い場合は処理終了する。

【0188】

6) 選択した再生手段で論理ページ毎に選んだ処理系でデスプールの処理を行い、物理ページ単位でページ排出を行う処理

1 物理ページの印刷に必要な、論理ページ全ての、処理系、処理階調数が揃ったところで、デスプールの処理を開始する。

【0189】

デスプーラは、最初にページの初期化命令を生成し、プリンタに送出する。

【0190】

後は、論理ページ毎にPDFファイルを読み込み、(5)で決定した処理系で描画命令の処理を行う。その際、レイアウト変換部(3034)で指定されたレイアウト変換処理を加えた描画命令を処理系に渡すことで描画位置の変更、拡大縮小処理といったレイアウト変換処理を実現している。

【0191】

次にサブクローズについて説明する。

【0192】

印刷命令のサイズが大きくて中間言語を1ページ分格納できなくなった時に、プリンタはサブクローズ状態となる。サブクローズ状態は、それまで蓄えた中間言語を、1ページ分のメモリ空間に描画処理を行うことで中間言語を格納していたメモリを開放し、新たに印刷命令を読み込むことを可能とするものである。このとき、プリンタの搭載メモリサイズが少ないと、指定階調数でサブクローズ処理が行えず、階調低下が発生する。こうした状態に入ると、プリンタの処理速度は極端に落ちる。そこでこれを予測し回避することが効果的な処理モード切替といえる。

【0193】

次にN-up印刷処理について説明する。

【0194】

N-u p 印刷とはアプリケーションソフトで作成された文書をプリンタドライバで1枚の用紙に何ページ印刷できるか指定できる機能であり、例えば1枚に4ページ印刷することを4-u pと呼ぶ。N-u p印刷は、物理ページ内のNページ分の変換処理が終わるまで、排紙コマンドを送らないことで、プリンタはNページ分の印刷命令を1ページ分の情報として処理することで実現している。

【0195】

論理ページ単位の処理系を決定した後の、N-u p印刷の処理の流れを図17のフローチャートを用いて説明する。なお、このフローチャートに係るプログラムはホストコンピュータ200のメモリに記憶され、CPUによって実行される。

【0196】

S1701: 次の論理ページの処理をすることに決定したトランスレータモジュールをロードし、S1702へ進む。

S1702: PDLモードならS1710へ、イメージモードならS1703へ進む。

S1703: 描画処理に用いるバンドメモリを獲得、ページ幅からバンド高さを算出し、バンドリストを生成する。S1704へ進む。

S1704: PDF(中間データ)をS1703で生成したバンドリストにリンクする。S1705へ進む。

S1705: レイアウト変換モジュールをこれから処理するバンド領域に合わせて初期化し、S12706へ進む。

S1706: 現バンド用バンドリストに繋がっている中間言語(PDF)を順にレイアウト変換モジュールに渡し、座標変換処理を行う。S1207へ進む。

S1707: 座標変換された描画命令を、トランスレータ(イメージモードレンダラー)に渡して、用意したバンドメモリ空間上に描画処理を実施する。S1708へ進む。

S1708: バンドリストにデータがまだ繋がっているならS1706へ、ないならS1709へ進む。

S 1 7 0 9 : バンドに描かれた結果からデバイスビットマップデータ (YMCK 各色 1, 2 o r 4 b i t イメージデータ) 生成、論理ページ位置から求めたレイアウト情報を加味して印刷命令生成し、S 1 7 1 0 へ進む。

S 1 7 1 0 : 全バンド処理終了なら S 1 7 1 5 へ、まだなら 1 7 0 5 へ進む。

S 1 7 1 1 : レイアウト変換モジュールを論理ページ領域に合わせて初期化し、S 1 7 1 2 へ進む。

S 1 7 1 2 : P D F 内の描画命令を順にレイアウト変換モジュールに渡し、座標変換処理実行する。S 1 2 1 3 へ進む。

S 1 2 1 3 : 座標変換が加えられた描画命令を、P D L トランスレータに渡し、印刷命令生成し、S 1 7 1 4 へ進む。

S 1 7 1 4 : P D F 内に描画命令がまだあれば、1 7 1 2 へ進む。

S 1 7 1 5 : 論理ページがまだあれば、S 1 7 0 1 へ、なければ本ジョブの処理終了する。

【0197】

次に、図 1 8 の模式図を用いて、N - u p 処理時の印刷命令生成処理を説明する。1 8 0 1 は図 1 8 でも説明に用いた高解像度イメージが貼り付けられた処理の重いページとする。本ページをイメージモードで処理する場合、描画処理用メモリ領域 1 8 0 4 を獲得し、これをバンドメモリ 1 8 0 5 として使うことでページ全面のメモリ空間を持たずに描画処理が可能である。

【0198】

等倍出力する場合は、4 M b y t e の描画用メモリで、R G B 2 4 B P P イメージレンダラーで A 4 ページの処理を行うためには 1 7 個の高さ b a n d H のバンド領域に分割される。

【0199】

1 8 0 3 : スプールされている P D F は、元々、描画原点ページの左上 (0, 0)、幅 W、高さ H のページに描画するように指示されたものである。これを、分割されたバンドメモリ領域に描画処理を行うためには、各バンド用バンドリストを用意し、それぞれのバンドに関わりのある P D F データを描画指定順に繋ぎなおす。生成したバンドリストに繋がっている描画命令を、バンド毎にトランスレ

ータ（レンダラー）に渡すことで描画処理を実現している。

【0200】

例えば、図18の1801の中のイメージデータは、左上描画位置（X1，Y1）、右下描画位置（X2，Y2）のイメージは、1803では2～7バンド目にかかっている。描画処理は、PDFに格納されている描画位置情報のY座標値Y1を、処理中のバンド位置内の座標系に変換したものをトランスレータ（レンダラー）にセットし、バンド領域でクリップ処理することで実現している。

【0201】

つまり、最初のバンドNoを0、現在処理しているバンドNoをbandNoとする時、

$$Y1' = Y1 - \text{bandH} * \text{bandno}$$

$$Y2' = Y2 - \text{bandH} * \text{bandno}$$

で示される。

【0202】

以上、生成されたピクセル辺りRGB24BPPのバンドメモリ空間に展開された出力結果は、バンド単位でYMCK各色8bitに色空間変換の後、ハーフトニング処理により出力階調数（各色1，2or4）に落とす。データにイメージ描画命令のコマンドを付加することで、このページの描画命令となる。

【0203】

次に、4-up出力時の印刷処理時について説明を行う。本ページの処理を行う前に既に文字のみのページが3ページ分PDLトランスレータによって処理が終了しているものとする。

【0204】

本ページをRGB24BPPイメージレンダラーで処理する場合、A4ページの1/4処理をRG24BPPレンダラーで行うためには、等倍時と同じ4Mbyteのメモリ領域で4個のバンドしか必要としない。

【0205】

スプールされているPDFは、元々、幅W、高さHのページに描画するように指示されたもののため、4-up印刷するためには、1304に示された幅、高

さを (w', h') の領域に領域に座標変換して描画処理を行わなくてはならない。等倍印刷時と同様に、各バンド用バンドリストを生成するが、その際に、PDF内の描画命令の座標値を描画先の座標空間に変換する。例えば1801の中のイメージデータの左上描画位置 $(X1, Y1)$ 、右下描画位置 $(X2, Y2)$ はそれぞれ、

$(X1' = X1 * (w' / W), Y1' = Y1 * (h' / H))$ 、 $(X2' = X2 * (w' / W), Y2' = Y2 * (h' / H))$ に変換される。

【0206】

描画処理は、バンドリストにつながれたデータの上記変換された描画位置のY座標値を、さらに処理中のバンド位置内の座標系に変換したものをトランスレータ（レンダラ）にセットし、バンド領域でクリップ処理することで実現している。

【0207】

以上、生成されたピクセル辺りRGB24BPPにバンドメモリ空間に展開された出力結果は、バンド単位でYMCK各色8bitに色空間変換の後、ハーフトーニング処理により出力階調数（各色1, 2 or 4）に落とす。生成したデータに4-up用描画位置に変換したイメージ描画命令のコマンドを付加することで、本論理ページ分の描画命令となる。

【0208】

4ページ分の描画命令が全て送出された後、排紙コマンドをプリンタに送出する。

【0209】

（プリンタ側のさらに詳しい説明）

本発明を実現するためには、プリンタは以下の機能が必要である。

【0210】

「デバイスビットマップ直通描画機能」

RGB色空間のデータは、プリンタ内部で色変換処理が加えられ、デバイス依存色空間データ（YMCK）となり、中間データとして保持される。ページメモリに描画時に、前記中間データはハーフトーニング処理がかけられ階調を落とし

たデバイスビットマップ形式となる。本発明は、PDLコマンドにデバイスビットマップデータを追加することで、論理ページ単位の処理モード切替を実現するものである。プリンタはデバイスビットマップデータを受け取った時に、そのまま中間データとして保持し、ハーフトーニング処理を加えずにページメモリに描画処理を行う処理系を持つ必要がある。

【0211】

次に、図19、20を用いて、モード選択の説明、及びそのフローの説明を行なう。

【0212】

図19はプリンタドライバのグラフィックモード設定画面である。

【0213】

1901のオートモードが指定されれば、図5、または図14の最適モード決定処理によりモードを決定し、アプリケーションからのデータを解析し、イメージデータまたはPDLデータに変換し、プリンタに転送する。

【0214】

1902のイメージモードが指定されていれば、イメージレンダラ3036を選択し、アプリケーションからのデータをイメージデータに変換し、プリンタに転送する。

【0215】

1903のPDLモードが指定されていれば、PDL3035を選択し、アプリケーションからのデータをPDLデータに変換し、プリンタに転送する。

【0216】

次に、図20を用いてグラフィックモード選択処理について説明する。なお、図20のフローにかかわるプログラムはホストコンピュータ200のメモリに記憶されており、CPUによって実行される。

【0217】

S2001でオートモードが指定されているか否かを判別し、指定されていればS2002に進み、指定されていなければS2003に進む。

【0218】

S 2 0 0 2では、図 5 または図 1 4 の処理によりアプリケーションからのデータを解析し、最適なレンダラを選択し、選択されたレンダラを用いてアプリケーションからのデータをイメージデータまたは P D L データに変換し、プリンタに転送する。

【 0 2 1 9】

S 2 0 0 4では、イメージモードが指定されているか否かを判別し、指定されていれば S 2 0 0 5に進み、指定されていないならば S 2 0 0 6に進む。

【 0 2 2 0】

S 2 0 0 5では、イメージレンダラ 3 0 3 6を選択し、選択されたレンダラを用いてアプリケーションからのデータをイメージデータに変換し、プリンタに転送する。

【 0 2 2 1】

S 2 0 0 6では、P D L モードが指定されているか否かを判別し、指定されていれば S 2 0 0 7に進み、指定されていないならば S 2 0 0 8に進む。

【 0 2 2 2】

S 2 0 0 7では、P D L レンダラ 3 0 3 5を選択し、選択されたレンダラを用いてアプリケーションからのデータを P D L データに変換し、プリンタに転送する。

【 0 2 2 3】

(第三の実施の形態)

以下に、本発明における第二の実施の形態を説明する。なお本実施の形態において第一の実施の形態と構成については、その説明を省略する。

【 0 2 2 4】

第一の実施の形態では、カラープリンタのカラーモードにおける各印刷モードの切替方法について説明していたが、本発明は、モノクロプリンタ、またはカラープリンタのモノクロモードにも適用可能である。以下にモノクロモード時の切替方法について差分部分の説明を行う。モノクロモード時は、以下のチェック項目

①カラーデータチェック

②データサイズチェック

③描画速度チェック

④論理演算値チェック

のうち、カラーデータチェックは行う必要はないが、それ以外は同様にチェックし、同様にモード切替の情報として利用する。

【0225】

但し、モノクロモードにおいては、データサイズチェックはプリンタ内部ではモノクロデータでデータを保持する場合は、そのサイズを用いる。

【0226】

また、②、③のチェックによりPDLモードで処理すると速度が遅くなる場合に移行するDeviceイメージモードはBlack1,2 or 4bitのイメージモードに対応する。

【0227】

④の論理演算値チェックで引っかかった場合に逃げるRGB24BPPイメージモードはやはりそのままである。

【0228】

(第四の実施の形態)

以下に、本発明における第四の実施の形態を説明する。なお本実施の形態において第一の実施の形態と構成については、その説明を省略する。

【0229】

第一の実施の形態では、PDLモード、Deviceイメージモード、RGB24BPPイメージモード全てが揃った場合についてであったが、何かがかけた場合でも、スプールしてチェックすることによって最も高速で最も正確な出力可能なモードに切り替えるものであってもよい。

【0230】

(その他の実施の形態)

なお、本発明は、複数の機器(例えばホストコンピュータ、インタフェイス機器、リーダ、プリンタなど)から構成されるシステムに適用しても、1つの機器からなる装置(例えば、複写機、ファクシミリ装置など)に適用しても良い。

【 0 2 3 1 】

また、本発明の目的は、前述した実施形態の機能を実現するプリンタドライバと呼ばれるソフトウェアプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ(またはCPUやMPU)が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、達成されることは言うまでもない。

【 0 2 3 2 】

この場合、記憶媒体から読み出されたプログラムコード事態が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【 0 2 3 3 】

プログラムコードを供給するための記憶媒体としては、例えば、フロッピーディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモリカード、ROMなどを用いることができる。

【 0 2 3 4 】

また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS(オペレーティングシステム)などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【 0 2 3 5 】

さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書き込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることはいうまでもない。

【 0 2 3 6 】

本発明を記憶媒体に適応する場合、その記憶媒体には、先に説明したフローチ

ャートに対応するプログラムコードを格納することになる。

【 0 2 3 7 】

以上説明したように、本発明の実施の形態によれば、プリンタドライバがOSから渡される印刷データを最低1ページ分保持する機構を持ち、保持する際に、印刷データのチェックを行い、そのページまたはバンドはどのモードで処理するのが最適かを判断し、決定したモードでページまたはバンド単位で処理し、高速に正確な描画処理を行うことを可能とする。

【 0 2 3 8 】

また、複数のモードで1頁を出力する際、選択されたモードで処理できる最大のバンド高さで処理を行うことにより、効率的に出力処理を行うことができる。

【 0 2 3 9 】

【発明の効果】

以上説明したように本発明によれば、印刷データに応じて最適な変換手段を選択することにより高速に印刷を行うことができる。

【 0 2 4 0 】

また、本発明によれば、N-u p 指定時でも、論理ページ単位で変換手段を選択するので高速に印刷を行うことを可能とする。

【図面の簡単な説明】

【図 1】

本発明の実施の形態が適用可能なレーザビームカラープリンタの構造を示す側断面図である。

【図 2】

従来の描画方法を示す概念図である。

【図 3】

本発明の実施の形態である印刷システム全体の基本構成を示すブロック図である。

【図 4】

各モードの特徴を示す図である。

【図 5】

本発明のスプール時の各オブジェクトのチェック方法の流れを示すフローチャートを示す図である。

【図 6】

本発明のスプール時のページの全オブジェクトチェック終了後の処理の流れを示すフローチャートを示す図である。

【図 7】

本発明のデスプール時のバンドリストの作り方を示す模式図である。

【図 8】

本発明のデスプール時のバンドリストの作り方の流れを示すフローチャートを示す図である。

【図 9】

P D F の構造を示す図である。

【図 1 0】

P C F の構造を示す図である。

【図 1 1】

スプール時にチェックした結果を保持する仮想バンドについて説明する模式図

【図 1 2】

イメージモード系処理と P D L モード系処理の処理時間の特徴を示す模式図。

【図 1 3】

本発明の物理ページ単位で見た、処理モード決定の流れを示すフローチャート

【図 1 4】

本発明の論理ページ単位で見た、処理モード決定の流れを示すフローチャート

【図 1 5】

P C F の構造を示す模式図。

【図 1 6】

本発明の理想的な実施例の模式図。

【図 1 7】

N - u p 時のデスプールの処理を示す図である。

【図 1 8】

N - u p 処理時の印刷命令生成処理を説明する図である。

【図 1 9】

グラフィックモード選択画面を示す図である。

【図 2 0】

グラフィックモード選択処理を示す図である。

【符号の説明】

- 1 0 0 印刷装置 (L B P)
- 1 1 0 フォーマッタ制御部
- 1 1 1 インタフェース
- 1 1 2 1 受信バッファ
- 1 1 2 2 送信バッファ
- 1 1 3 P D L 解析部
- 1 1 4 印刷制御処理実行部
- 1 1 5 描画処理実行部 (レンダラ)
- 1 1 6 ページメモリ
- 1 2 0 オペレーション・パネル
- 1 3 0 出力制御部
- 1 3 1 レーザドライバ
- 1 4 0 プリンタエンジン部
- 1 4 1 半導体レーザー
- 1 4 2 レーザ光
- 1 4 3 ポリゴンミラー
- 1 4 4 静電ドラム
- 1 4 5 現像ユニット
- 1 4 6 給紙カセット
- 1 4 7 給紙ローラ



- 148 搬送ローラ
- 149 搬送ローラ
- 200 ホスト・コンピュータ
- 201 アプリケーション・ソフト
- 202 グラフィック・サブ・システム
- 2021 GDI
- 2022 プリンタ・ドライバ
- 2023 PDLトランスレータ
- 2024 モノクロ1BPPイメージモード
- 2025 YMCK4BPPイメージモード
- 2026 RGB24BPPイメージモード
- 2027 イメージモード用バンドメモリ領域
- 2028 GDI
- 2029 インターフェース
- 210 キーボード
- 211 マウス
- 220 ディスプレイ・モニタ
- 300 ホスト・コンピュータ
- 301 アプリケーション・ソフト
- 302 グラフィック・サブ・システム
- 3021 GDI
- 3022 プリンタ・ドライバ
- 3023 プレチェックシステム
- 303 スプールサブシステム
- 3031 PDF (プリントデータファイル)
- 3032 PCF (プレチェックファイル)
- 3033 デスプーラー
- 3034 PDLトランスレータ
- 3035 モノクロ1BPPイメージモード

- 3036 YMCK4BPPイメージモード
- 3037 RGB24BPPイメージモード
- 3038 イメージモード用バンドメモリ領域
- 3039 インターフェース
- 210 キーボード
- 211 マウス
- 220 ディスプレイ・モニタ
- 900 PDF (ページデータファイル) の構造を示す模式図
- 901 pageID (ページを特定する番号)
- 902 PageInfo (ページサイズ、設定モードといった描画処理に必要な設定)
- 903 描画関数指定データ (DrvCopyBitsを指定)
- 904 dataSize (本描画データに掛かっているスプールサイズ)
- 905 soDest (psDestから獲得した情報)
- 906 soSrc (psSrcから獲得した情報)
- 907 ソースイメージデータ (psSrcが示すソースイメージデータ本体)
- 908 co (pcoが示すクリップ情報)
- 909 クリップデータ (クリップ処理に必要なデータ)
- 910 xlo (pxloが示す色変換情報)
- 911 rclDest (prclDestが示す描画先矩形位置情報)
- 912 ptlSrc (ソースデータの位置を示す情報)
- 913 描画関数指定データ (DrvTextOutを指定)
- 1000 PCF (プレチェックファイル) の構造を示す模式図
- 1001 pageID (ページを特定する番号)
- 1002 dataSize (本ページのPCFのサイズ)
- 1003 setColMode (本ページの指定されたカラーモード)
- 1004 desiColMode (本ページの決定カラーモード)
- 1005 desiMode (本ページのモード)

1006 PDLDataSize (PDLデータで送った場合のプリンタ内部のデータサイズ予測値)

1007 ImageDataSize (イメージモードで送った場合のデータサイズ)

1008 bandCnt (チェック用バンドの数)

1009 チェック用バンドの情報構造体

10091 bandNo (バンドナンバー)

10092 desiMode (決定した処理モード)

10093 rendTime (本バンドをPDLモードで処理した場合の処理時間予測値)

10094 BandRectL (本バンドの領域)

10095 BandRendRectL (本バンドの描画領域の際外郭矩形)

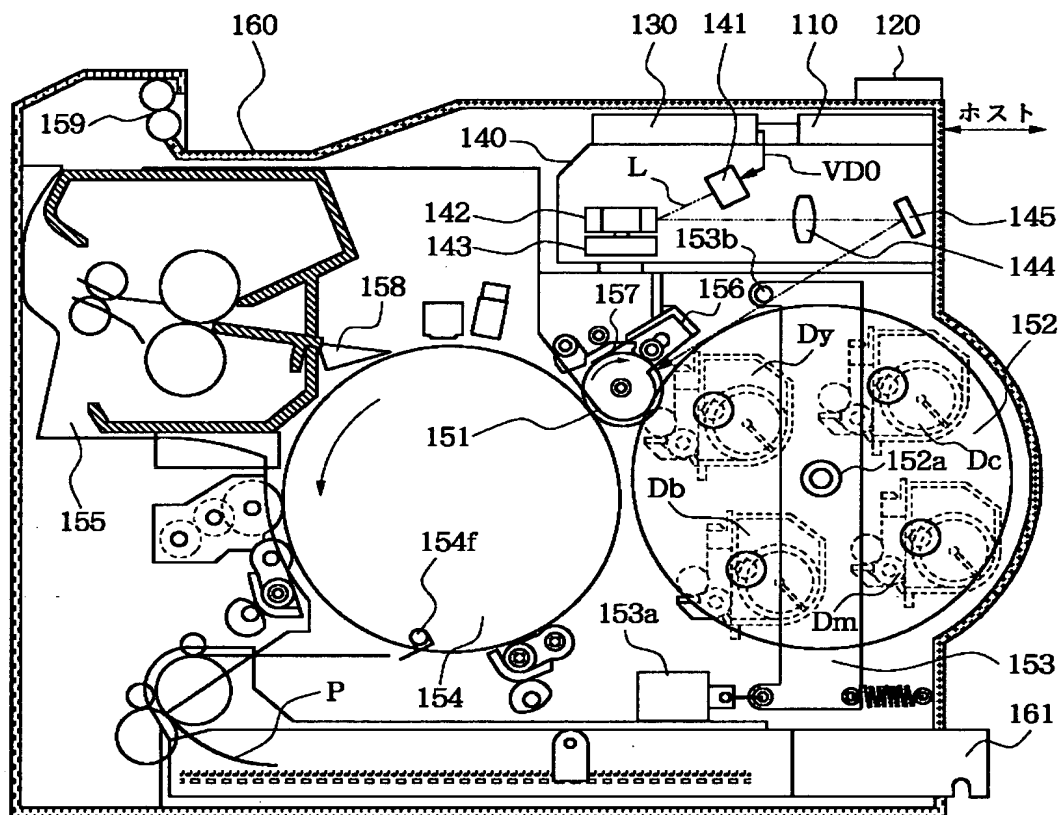
10096 PDLDataSize (PDLモードで処理した場合の本バンド描画に掛かるデータ数)

10097 objflg (描画オブジェクトが一つでもあればフラグを立てる)

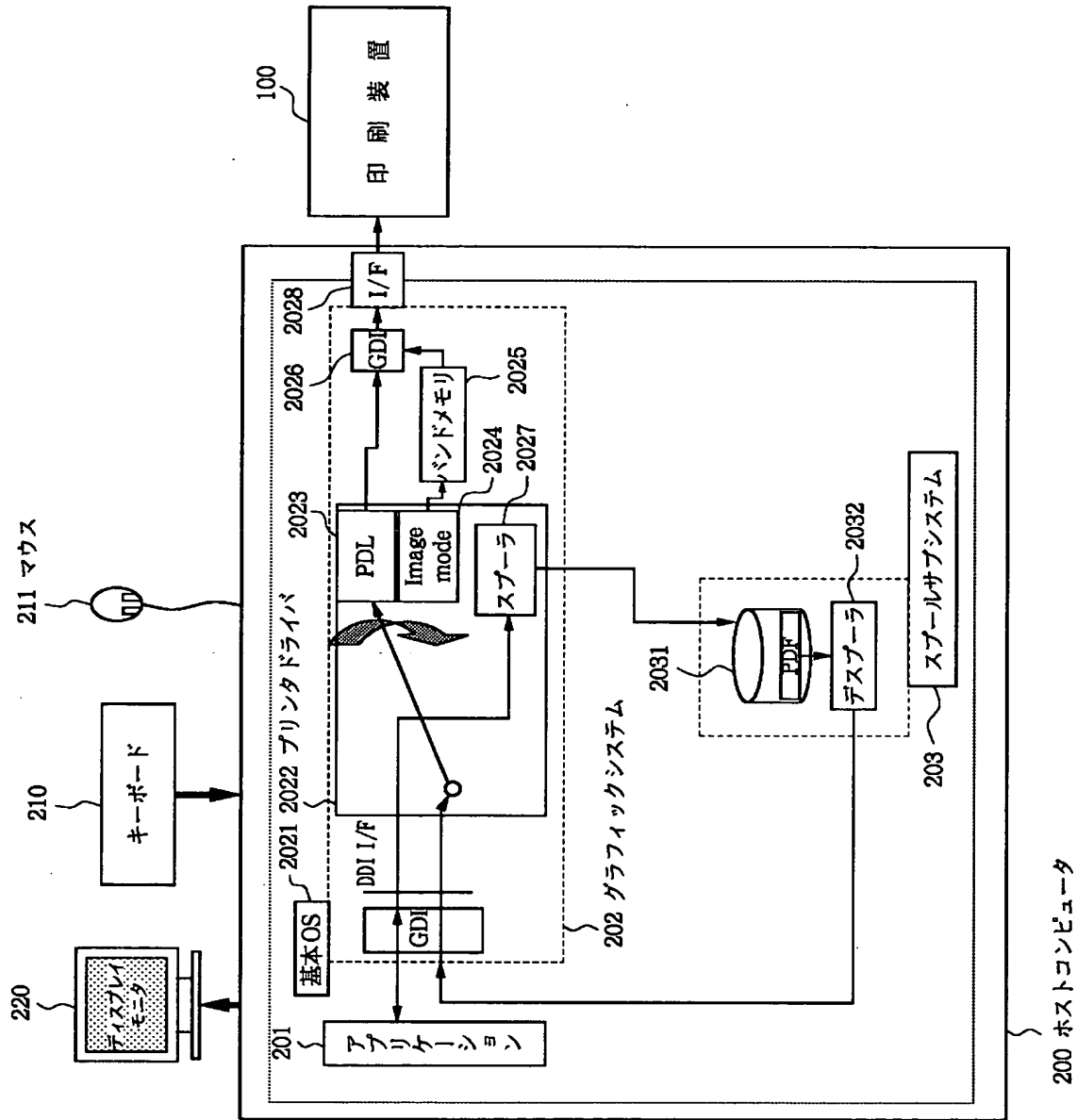
10098 colflg (描画データに有彩色が含まれる時はフラグを立てる)

【書類名】 図面

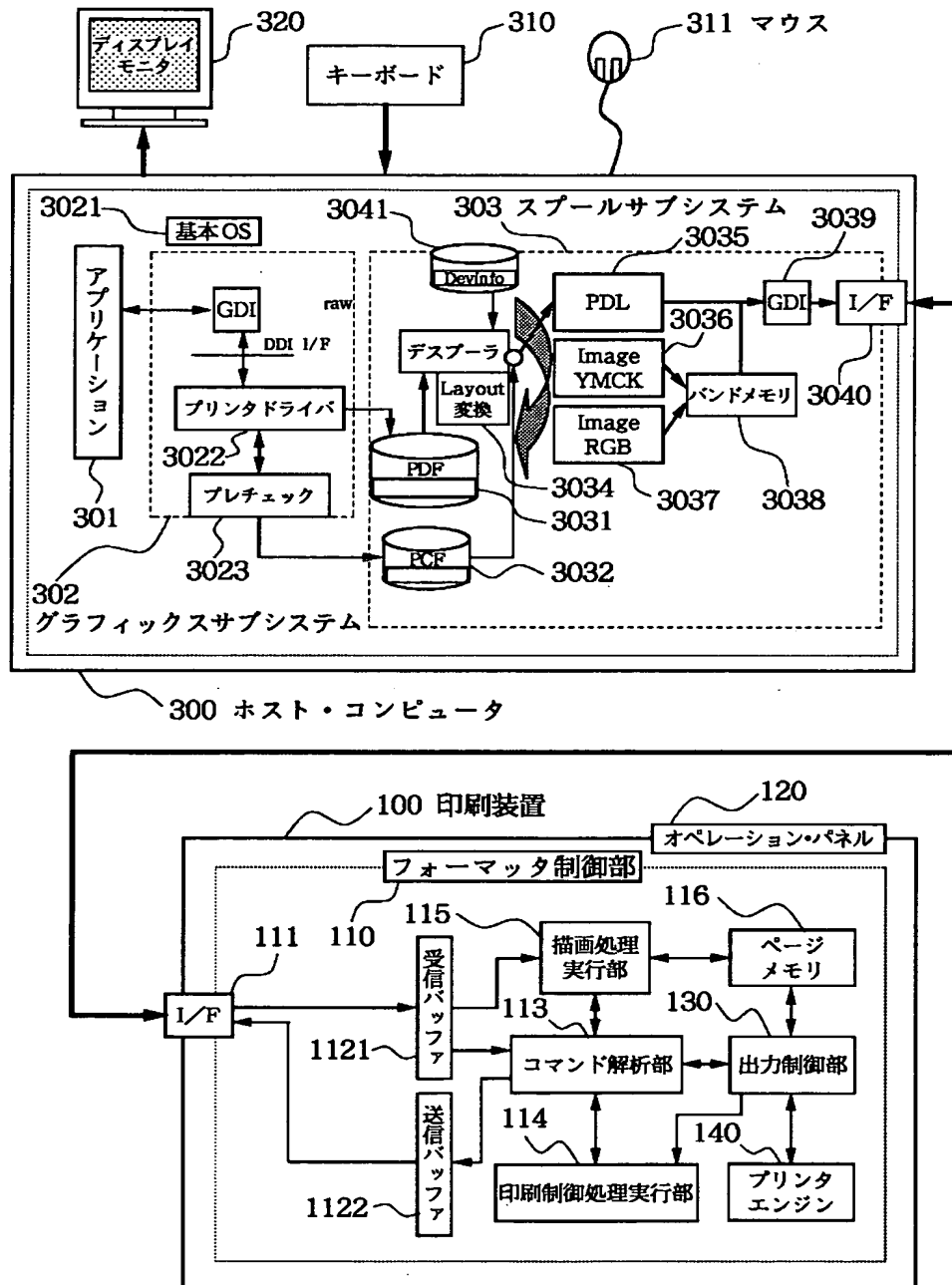
【図 1】



【図 2】



【図3】

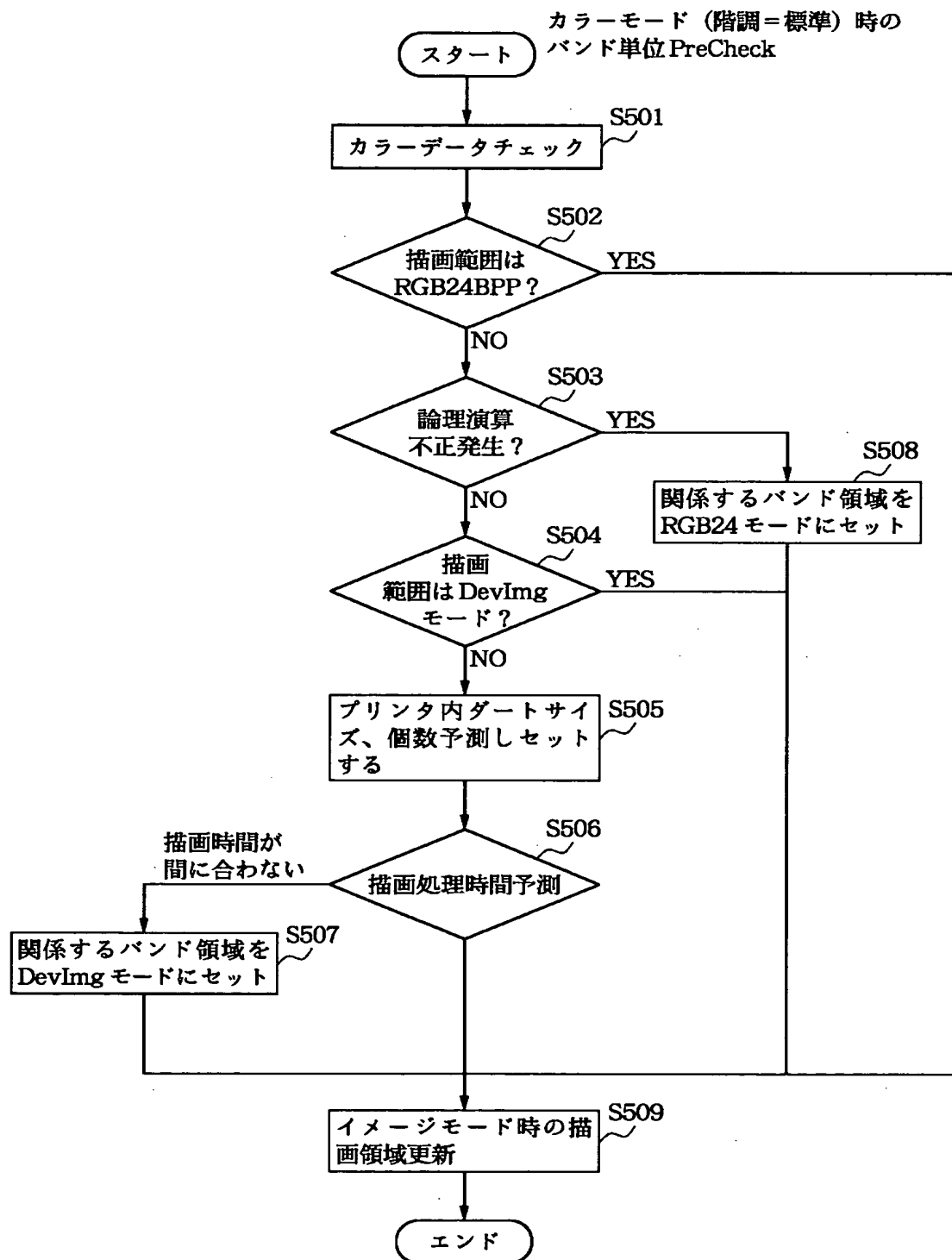


【図 4】

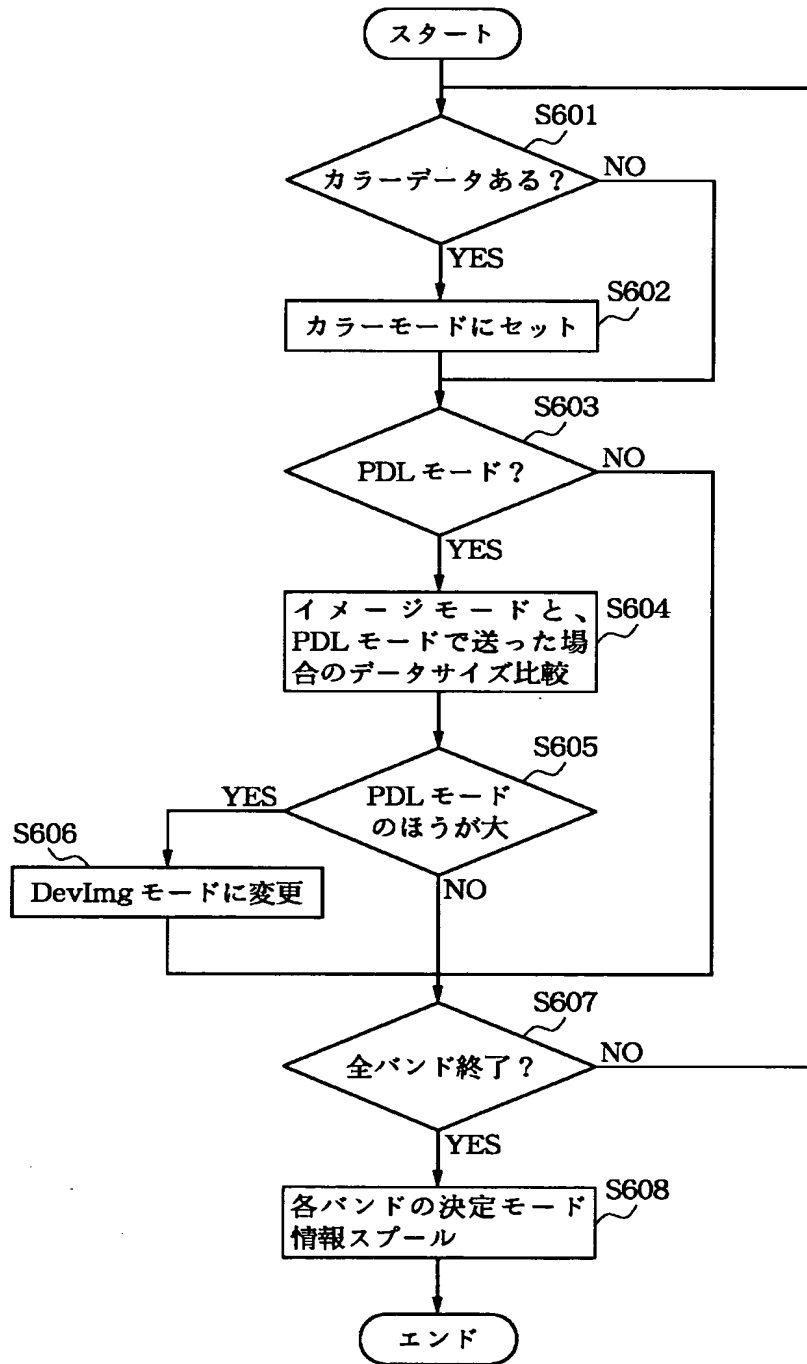
各処理モードの特徴

	データサイズ	処理速度	他の特徴
PDL	通常ページは最小 大きさに制限なし。	通常ページは高速だが 複雑なグラフィックス が低速	論理演算処理に問題 あり、
Device Image	描画面積に比例 最大値は固定	高速だが、簡単な図形 文字で、エンジン速度 が出ない	論理演算処理に問題 あり、
RGB24 Image	描画面積に比例 最大値は固定	低速	論理演算処理可能

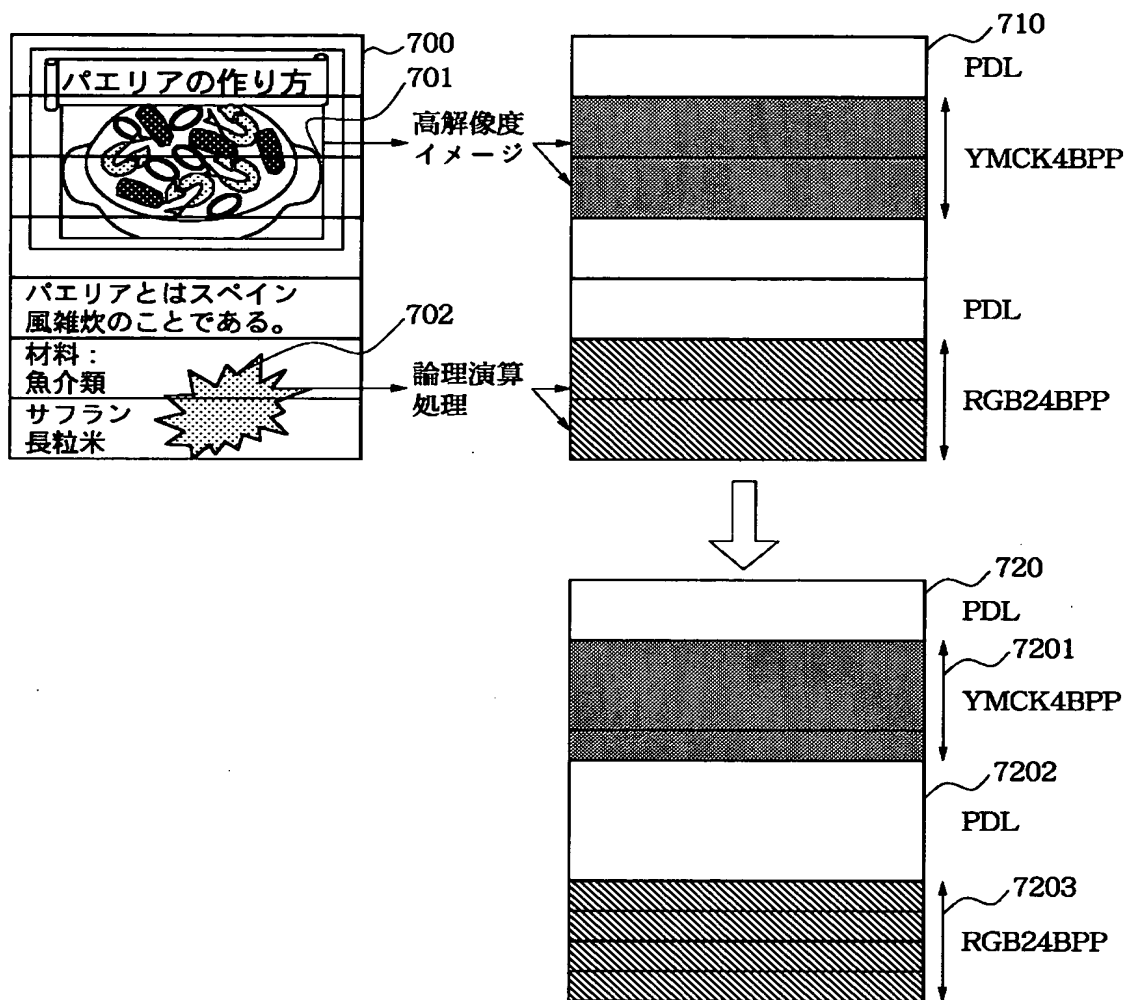
【図 5】



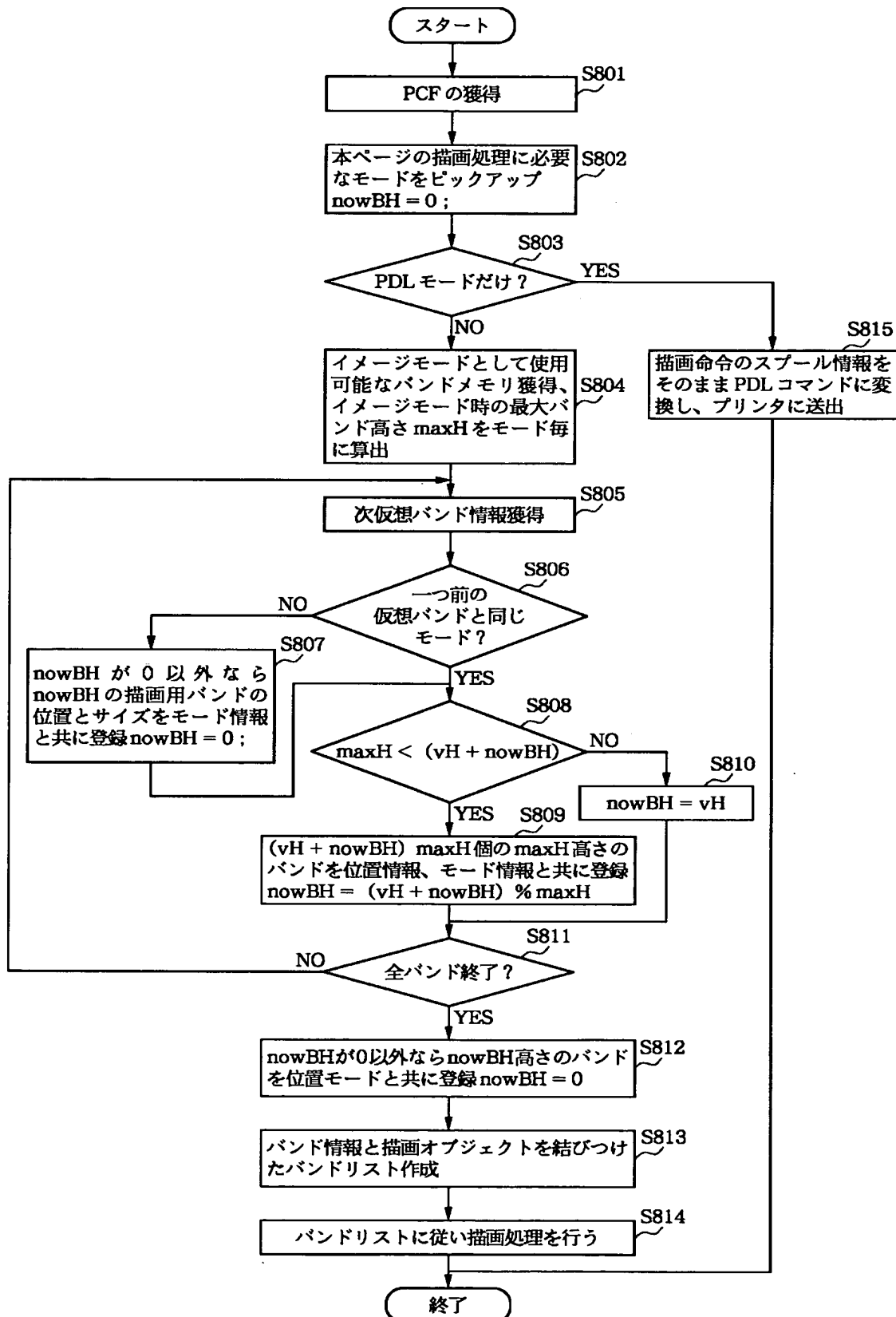
【図 6】



【図 7】

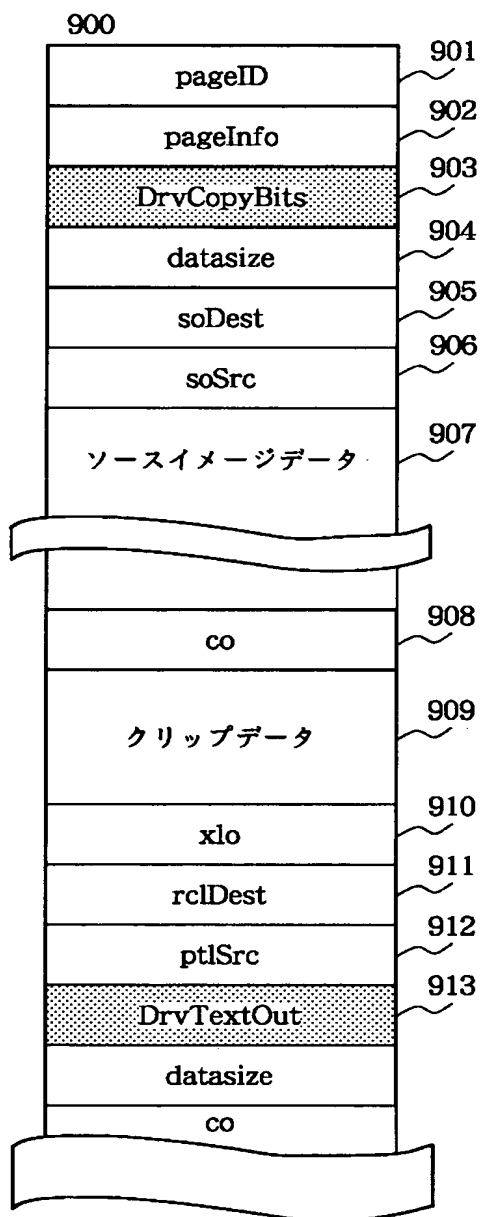


【図 8】



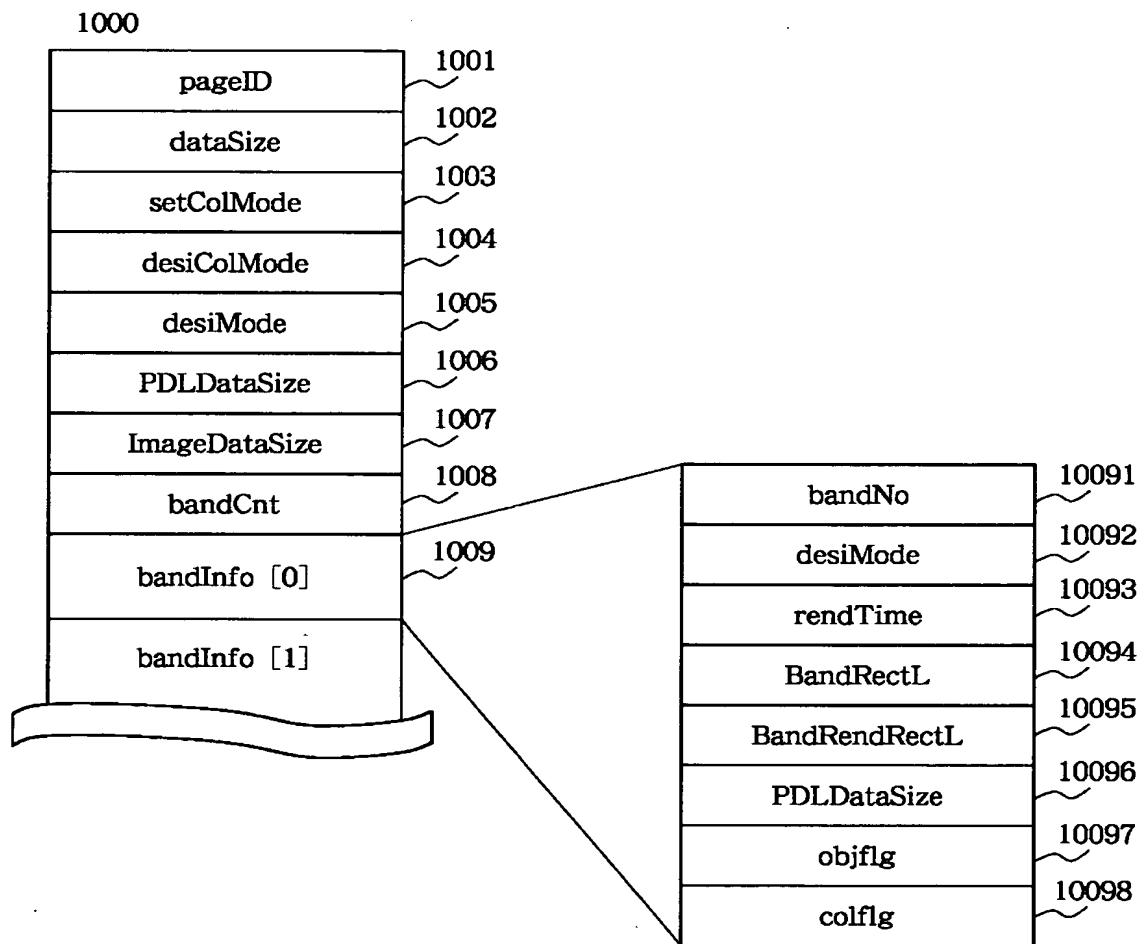
【図 9】

PDF の構造を示す模式図

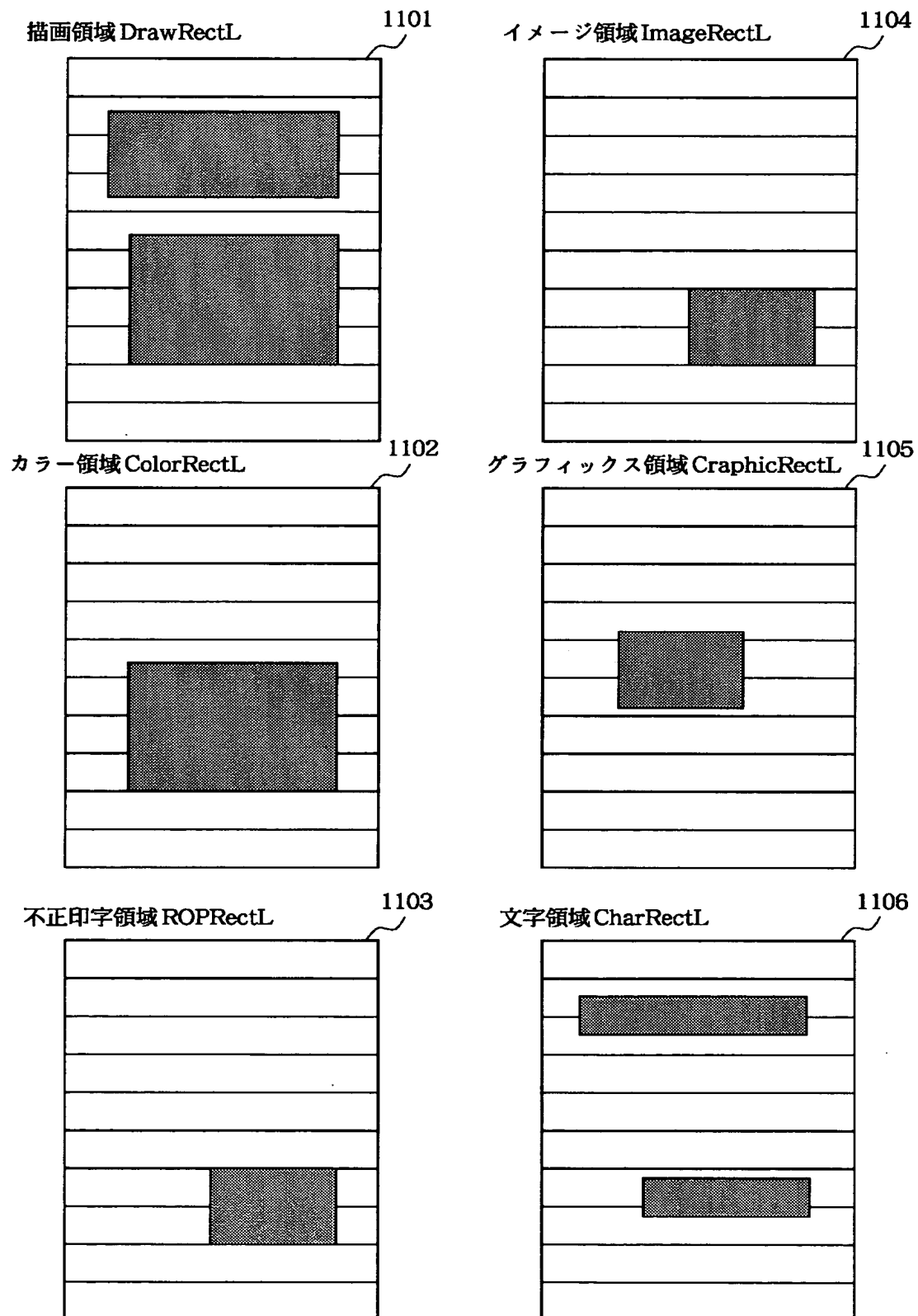


【図 1 0】

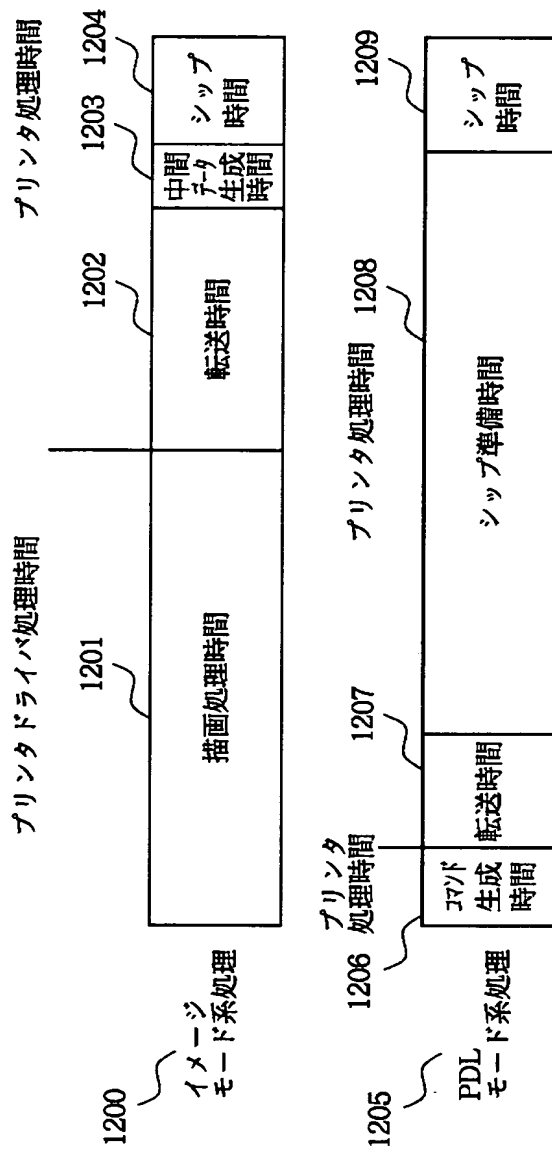
PCF の構造を示す模式図



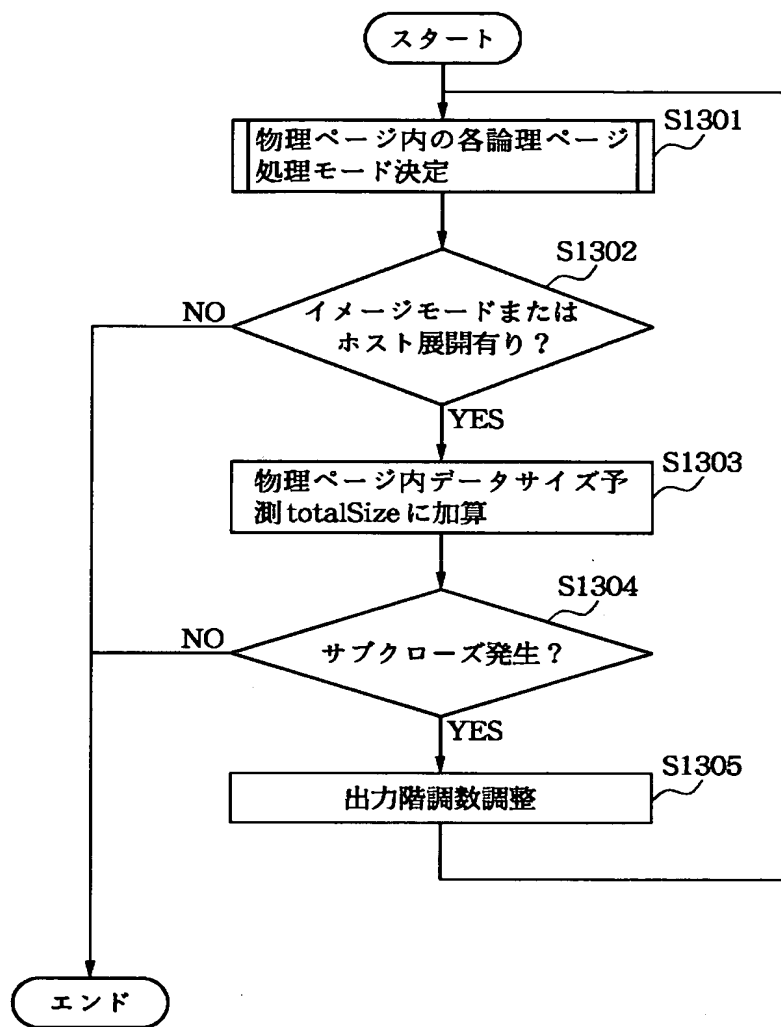
【図 1 1】



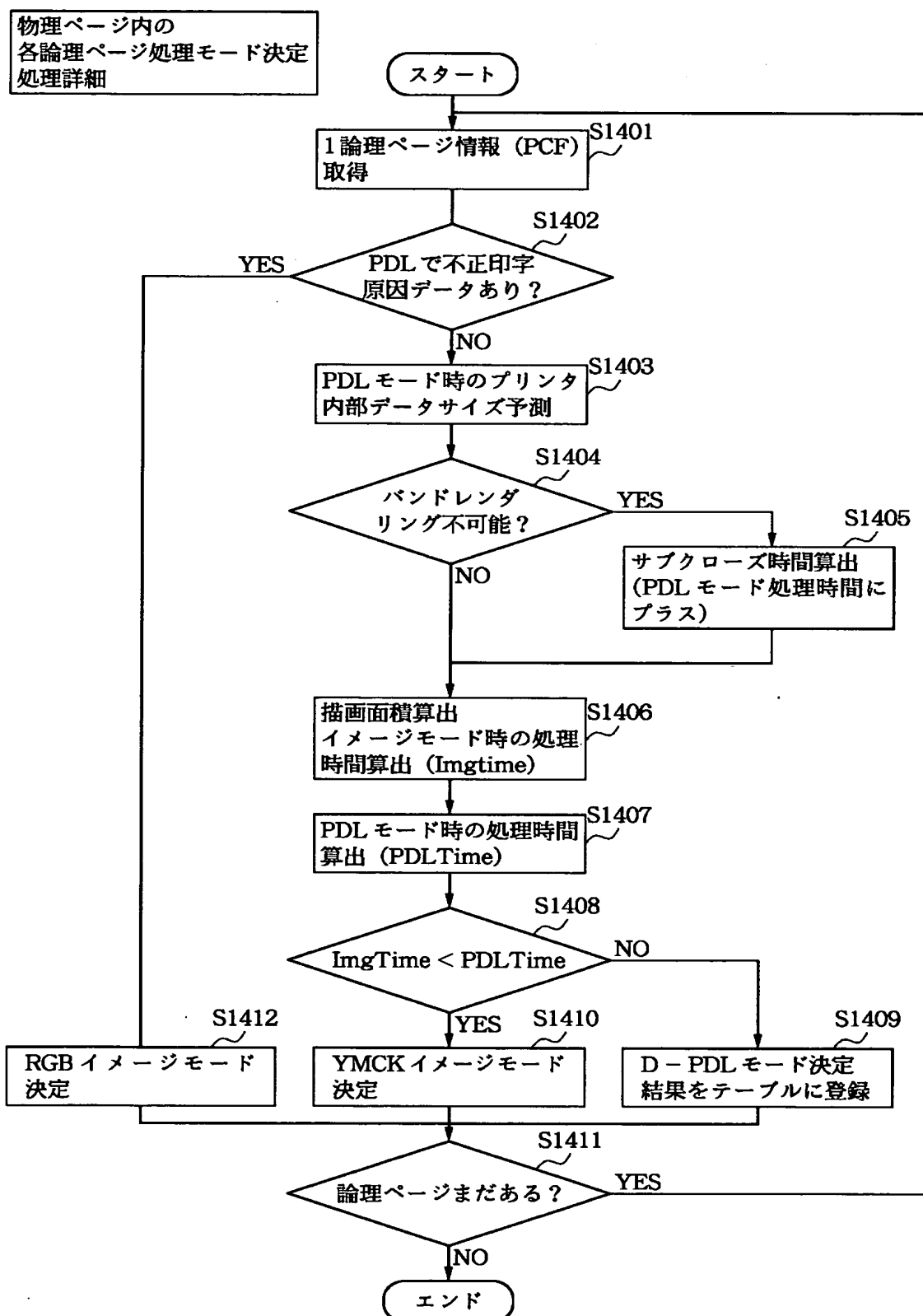
【図 1 2】



【図 1 3】

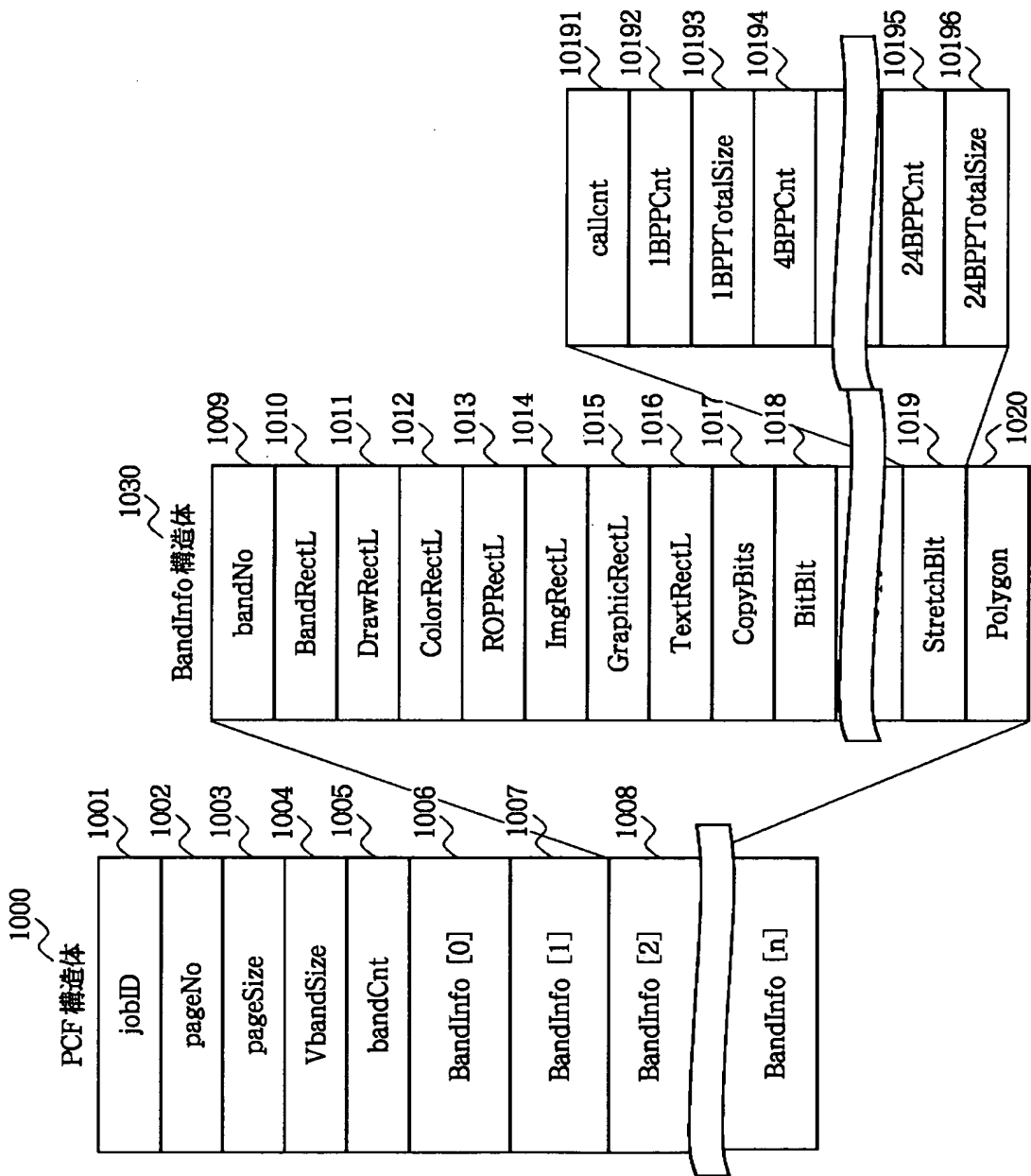


【図 14】

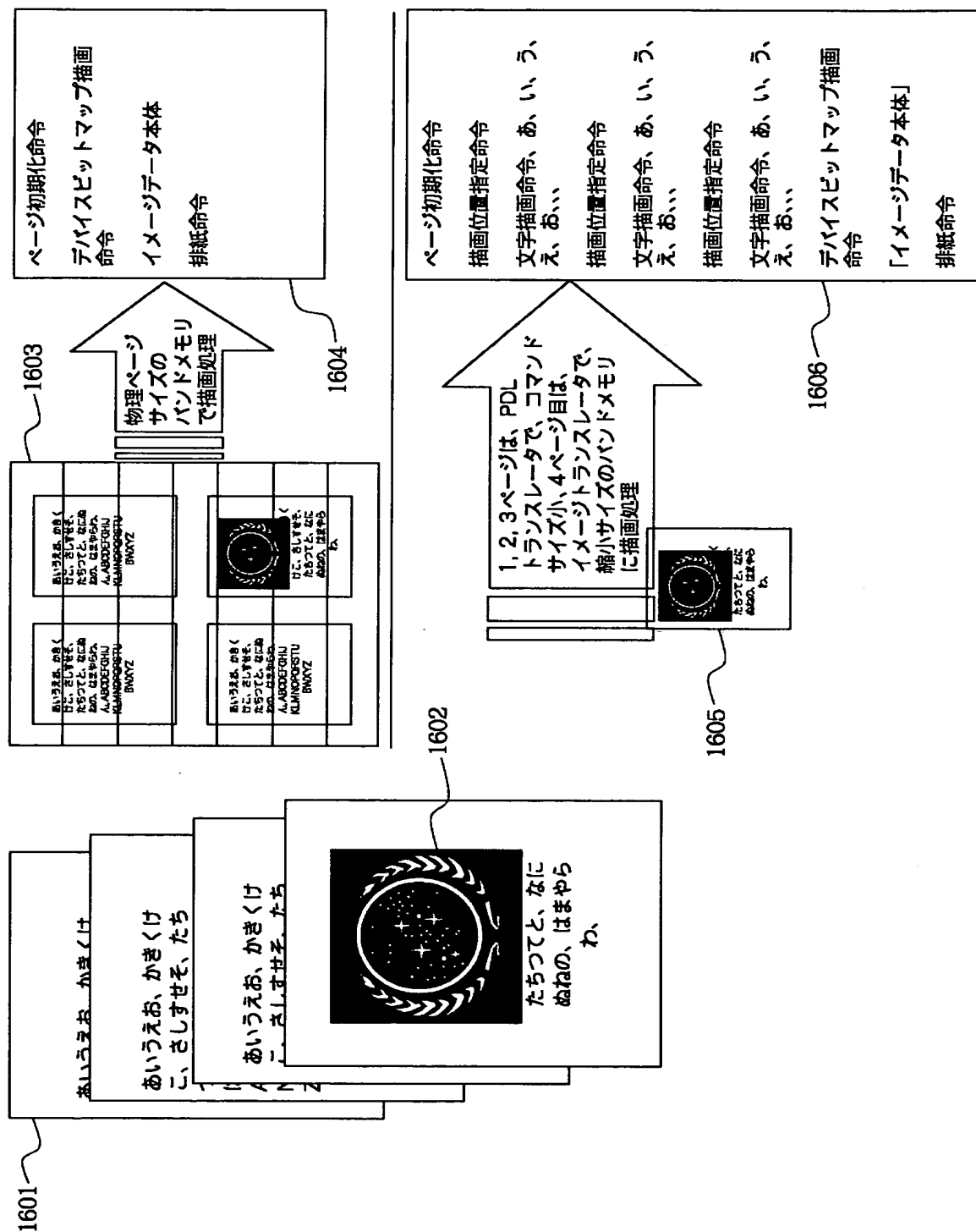


【 図 1 5 】

PCF の構造を示す模式図

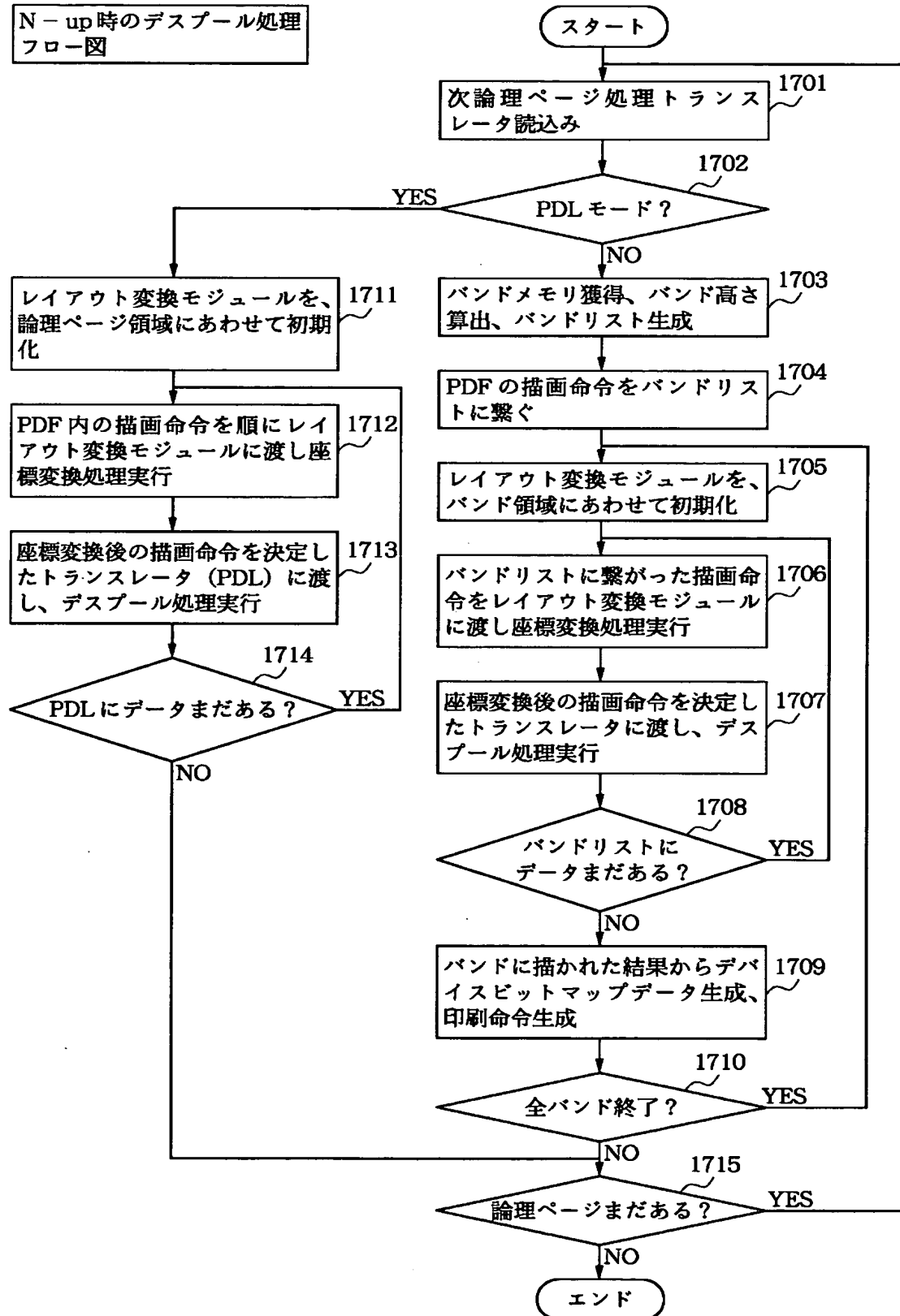


【図 16】

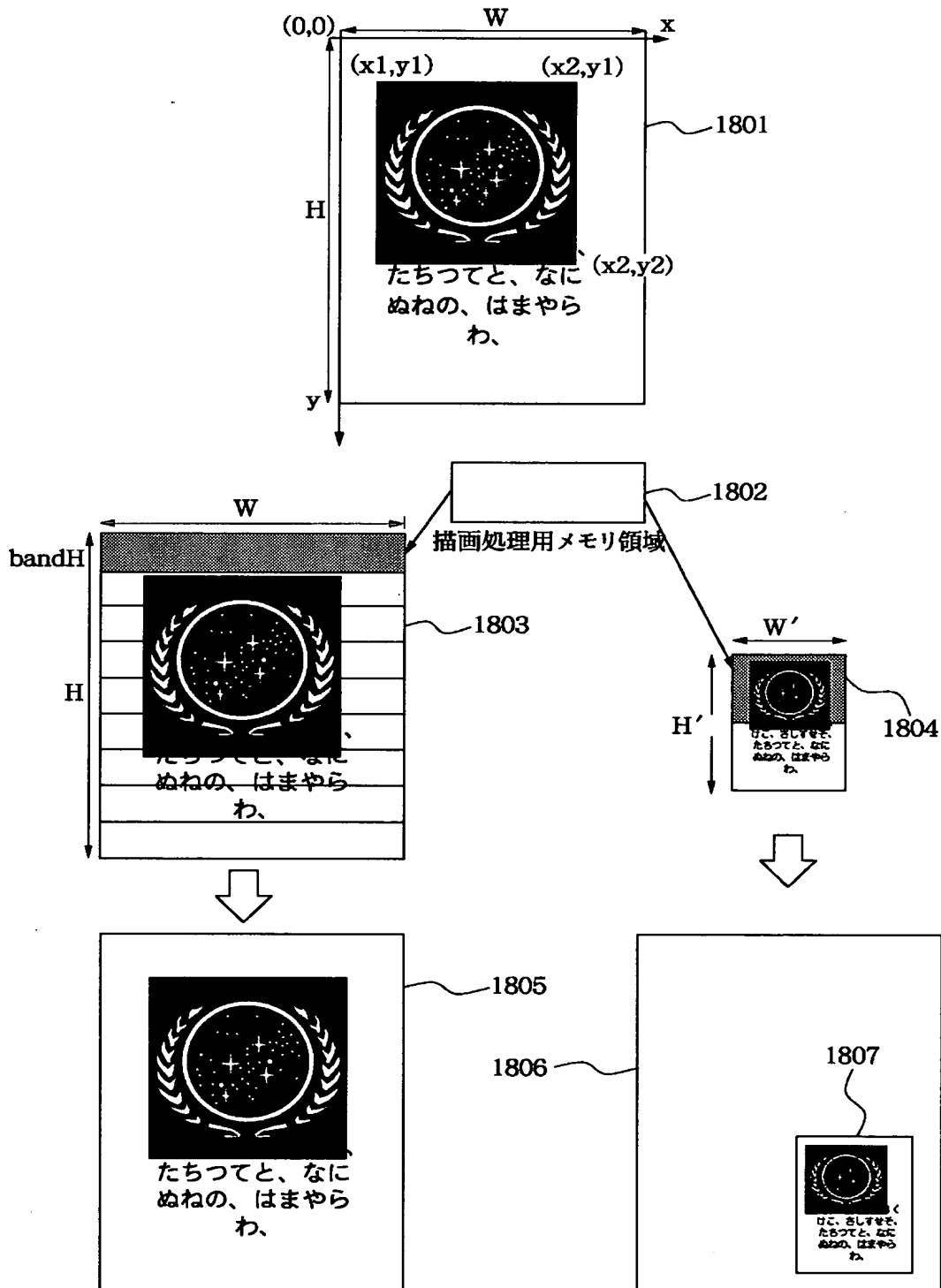


【図 17】

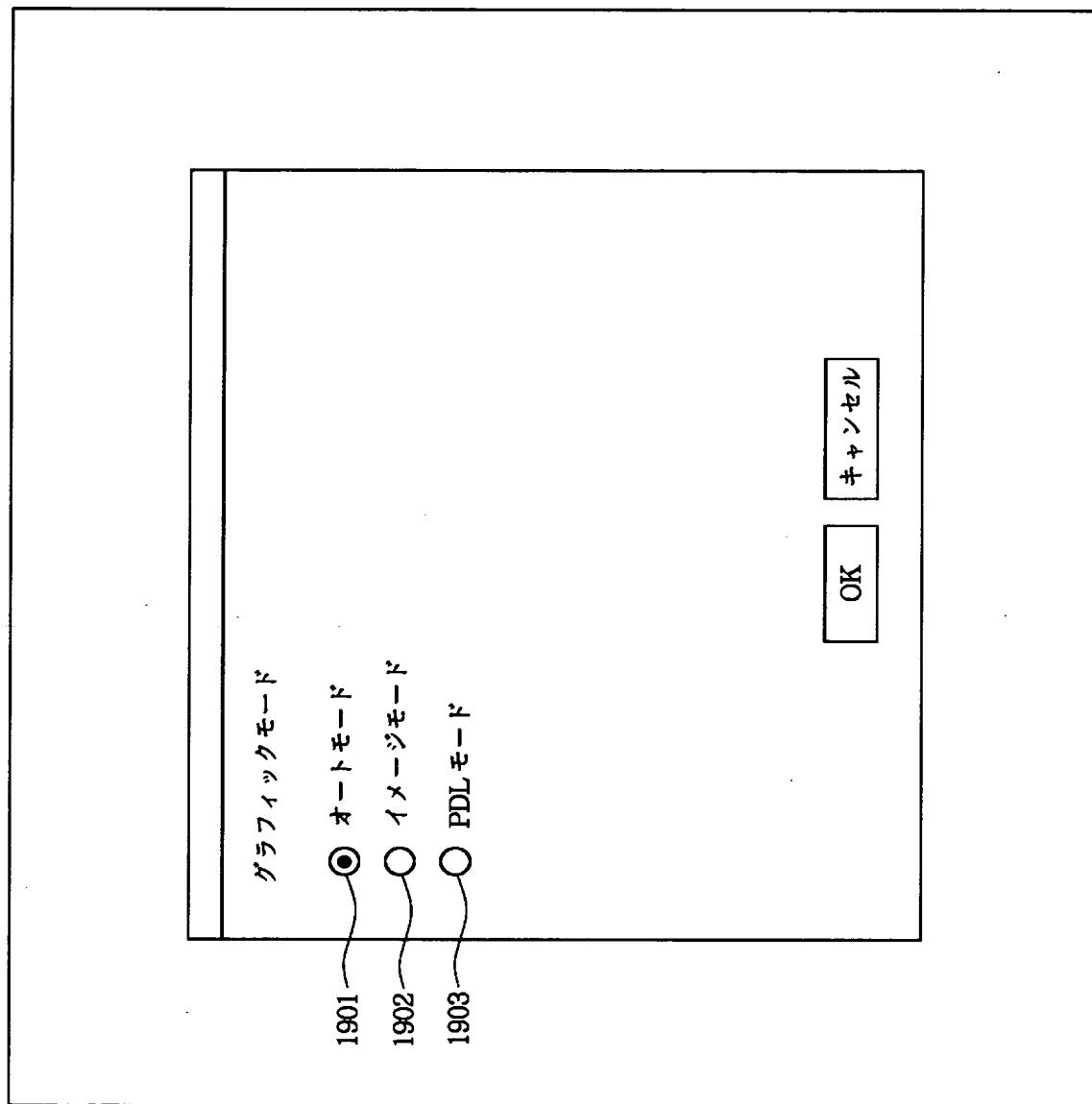
N-up時のデスプールの処理
フロー図



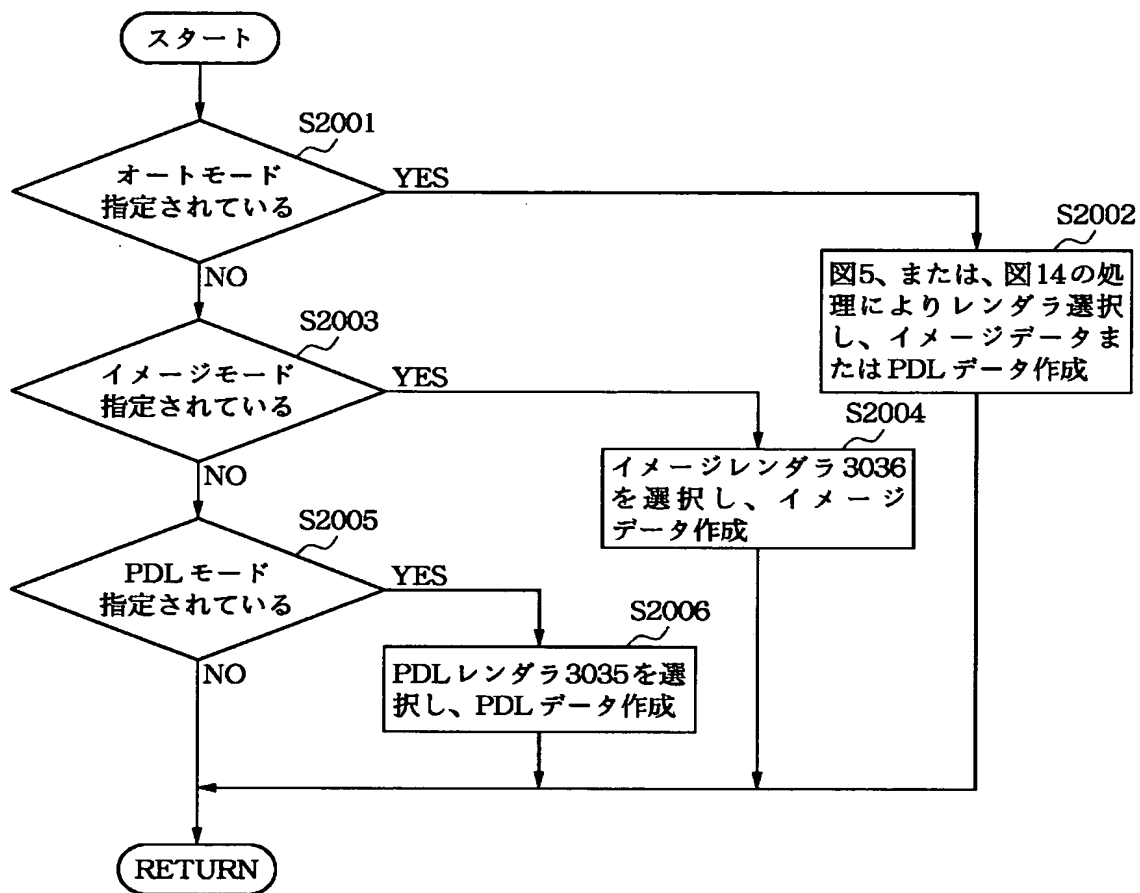
【図 18】



【図 1 9】



【図 2 0】



【書類名】 要約書

【要約】

【課題】 印刷データに応じて、最適な変換手段を選択し高速に印刷を行なうこと。

【解決手段】 印刷データをイメージデータに変換する第 1 変換手段と、
印刷データをプリンタでイメージデータを生成するコードデータに変換する第 2 変換手段と、

前記第 1 変換手段により変換させたイメージデータを用いた印刷時間と前記第 2 変換手段により変換させたコードデータを用いた印刷時間とを予測する予測手段と、

前記予測手段による予測結果に基づきイメージデータを用いた印刷時間の方が短いかな否かを判別する判別手段と、

前記判別手段によりイメージデータを用いた印刷時間の方が短いと判別された場合、前記第 1 変換手段を選択し、前記判別手段によりイメージデータを用いた印刷時間の方が短くないと判別された場合、前記第 2 変換手段を選択する選択手段とを有することを特徴とする。

【選択図】 図 1 4

認定・付加情報

特許出願の番号	特願 2000-170657
受付番号	50000706625
書類名	特許願
担当官	第七担当上席 0096
作成日	平成12年 6月12日

<認定情報・付加情報>

【特許出願人】

【識別番号】	000001007
【住所又は居所】	東京都大田区下丸子3丁目30番2号
【氏名又は名称】	キャノン株式会社

【代理人】

申請人

【識別番号】	100090538
【住所又は居所】	東京都大田区下丸子3丁目30番2号 キャノン株式会社内
【氏名又は名称】	西山 恵三

【選任した代理人】

【識別番号】	100096965
【住所又は居所】	東京都大田区下丸子3丁目30番2号 キャノン株式会社内
【氏名又は名称】	内尾 裕一

【選任した代理人】

【識別番号】	100110009
【住所又は居所】	東京都大田区下丸子3丁目30番2号 キャノン株式会社内
【氏名又は名称】	青木 康

出 願 人 履 歴 情 報

識別番号 [000001007]

1. 変更年月日 1990年 8月30日
[変更理由] 新規登録
住 所 東京都大田区下丸子3丁目30番2号
氏 名 キヤノン株式会社